Data Analysis of Low SNR signals in fast High-Noise Acquisition Systems and Validation with Miniaturised Electrophoresis

by

Nicolaas Ockert Bester

B.Sc (Hons) (Natal); B.Mus (Hons) (Brus); M.Sc (York)

Australian Centre for Research on Separation Science (ACROSS)

College of Sciences and Engineering

Submitted in fulfilment of the requirements for the Degree of

Doctor of Philosophy

University of Tasmania October, 2018

Declaration

To the best of my knowledge, this thesis contains no copy or paraphrase of material previously published or written by another person, except where due reference is made in the text of the thesis.

Nicolaas Ockert Bester

15 October 2018

Nicolaas Ockert Bester

October 2018

DEDICATION

In Nomine Deo

and

in gratitude to my dear brother Paul, for his unfailing
love through difficult times,

and

to our mother and father,
whose selfless and tangible encouragement from an early age
instilled in both of us a love of knowledge, truth, beauty and learning.

ACKNOWLEDGEMENTS

Throughout my candidature I have been carefully supervised and mentored by my supervisors, Prof. Michael Breadmore and Prof. Rosanne Guijt. During the first part of my candidature, Prof. Guijt was able to provide me with valuable initial direction to publications on miniaturisation, and detailed guidance on calibration and acquisition of performance data during the building of my miniaturised apparatus. After Professor Guijt's departure to Germany and latterly her appointment to a senior position at another university, Prof. Breadmore shouldered the responsibility of face-to-face mentoring and supervision. During this time, Prof. Breadmore introduced me to two postdoctoral assistants, and as such I was fortunate to have the support and very useful input from both Dr. Hong Heng See and Dr. Min Zhang with regard to instrumental refinement and fast data acquisition respectively.

During all the difficulties which I experienced over the years of this candidature, both in health and also legal and social difficulties, Prof. Michael Breadmore has been both encouraging and supportive in every way. He has been equally uncompromising in his requirement for rigour, detail and substantive argument in all matters academic. I have always found our meetings to be challenging, exciting and stimulating – especially during those times where I was told in no uncertain terms that I had completely failed to address or grasp an essential concept; only from such truths can re-calibration occur and progress be made. The reductionism of the scientific method still remains the best means of engaging with nature, and the reasons for Prof. Breadmore's stature in the scientific community became clear to me as I engaged with him and struggled my way through some of the academic papers which form his *operis corporis*.

I would also like to register my gratitude and thanks for the invaluable help of members of the academic and technical staff; particularly Dr Murray Frith, Chemistry Laboratory Manager, Dr. Petr Smejkal for his assistance with the benchtop models of the Agilent CE apparatus, A/Prof. Ashley Townsend of the Central Science Laboratory for his benchmarking of data with ICP-MS and Mr. Paul Waller whose encyclopaedic knowledge of electronics and control systems was invaluable.

During my time in the student area of ACROSS, I was able to enjoy the friendship and companionship of my fellow students, especially Chowdhury Hasan, Ibraam Mikhail, Chris Desire, Umme Kalsoom, Farhan Cecil and Brenda Mooney.

Finally I would like to register my thanks for the kindness and support of three stalwart ladies of great integrity: Dr. Ruth Amos, Ms Carol Jacobs, and the unflappable, all-knowing, and frighteningly efficient Patricia McKay.

List of Abbreviations and Definitions

ADC	Analogue to Digital Converter
AMD	Acidic Mining Drainage
AR	Analytical Reagent
ARIMA	Autoregressive Integrated Moving Average (see EMS)
baud	Unit of transmission speed equal to the number of times a signal changes state per second. For signals with only two possible states one baud is equivalent to one bit per second.
BGE	Background Electrolyte
bit rate	The bit rate R is given by: $R = \text{baud rate} \times \log_2 S = \text{baud rate} \times 3.32 \log_{10} S$
C++	"C" programming language with two additional classes
CD	Chemiluminescence Detection
CE	Capillary Electrophoresis
CMS	A content management system (CMS) is a software application or set of related programs that are used to create and manage digital signals.
.csv	comma separated text file
DAQ	Data Acquisition
DIS	Data-intrusive smoothing. A hardware data-smoothing device such as a RC circuit which is inserted between detector and receiving instrument.
DC	Direct Current
DWA	Discrete Wavelet Transform
EMS	Exponential Mean Smoothing

EWMA	Exponentially Weighted Moving Average (see EMS)
Fast DAQ	Data acquired at rates greater than 300Hz but less than 5 kHz
FFT	Fast Fourier Transform
FIA	Flow Injection Analysis
FITC	Fluorine Isothiocyanate
FT	Fourier Transform
GMS	Geometric Mean Smoothing
i.d.	internal diameter
LED	Light Emitting Diode
lemma (pl: lemmata)	A lemma is a necessary preliminary proof before the principal proof in a mathematical argument. A lemma should be much shorter than the main proof.
NRZ	Non-Return to Zero (NRZ) code is just a simple square wave, assigning one value to a binary 1, and another amplitude to a binary 0. In this case these are voltage values sampled from the photodiode by the Arduino board.
NSA	Nyquist Smoothing Algorithm
o.d.	outside diameter
PAR	4 - (pyridylazo) resorcinol
PD	Potential difference (Applied voltage)
PEEK	polyether ether ketone
PMP	Programmable Micro-Processor
PMT	Photomultiplier Tube
PWM	Pulse Width Modulation

RC	Resistor-Capacitor circuit which has a dampening effect on an incoming signal.
RSD	Relative Standard Deviation
S-G	Savitsky-Golay
SI-CE	Sequential Injection – Capillary Electrophoresis
Slow DAQ	Data acquired at rates below 300 Hz
Very Fast DAQ	Data acquired at rates greater than 5 kHz
$\chi_{(f)}$	Chi function
µTAS	Miniaturised Total Analysis System
Ξ Operator (Mathematical)	A process Ξ to be applied to a set of data points to achieve an outcome. A single operator may give more than one solution, and an operator is not necessarily a function.

Abstract

This thesis describes three areas of work and study; the first being experimental and instrumental, and the second two being the deriving of mathematical and numerical processes. All three areas deal with the vexed question of low concentration analyte signals within high noise backgrounds.

The first area of work deals with the design and construction of a small-scale compact sequential injection – capillary electrophoresis (SI-CE) apparatus to detect low concentrations (<100 ppb) of heavy metals. Various configurations of inexpensive components included small-scale pumps, valves, switches, miniaturised 3 kV HV units, monochromatic LEDs, and a 10-bit Arduino UNO™ control system were tested. In its final configuration, the instrument occupied a volume of 20×13×10 cm^3, with mass approximately 900 g. This instrument cost less than $AU1000, and was able to operate independently on a drill battery for three days. The battery-powered operation (n=1163 runs) showed consistent migration times and areas with RSDs < 2.0% and 9.0%, respectively. It was then shown to be able to detect lead levels in the low (<10 ppb) range, by pre-capillary complexation with PAR. The instrument was then calibrated against commercial (Agilent) instrumentation by separation of selected fluorophores.

During the construction and trial of this instrument, it became clear that a 10 bit ADC gave a very noisy and rough signal particularly at low concentration, and in order to reduce noise it was necessary to introduce smoothing software into the real-time data processing of the ADC. This problem lead to noise reduction by means of oversampling, using a novel application of the Nyquist Theorem and using a snapshot approach to smoothing. Without this Nyquist smoothing approach, the instrument would not have been of much use without using higher-end components, leading to considerable increase in cost, size and complexity.

The second body of work originated from ideas which come from smoothing of signals in the small-scale instrument. It is a comparative study of six different smoothing algorithms applied *post-facto* to unsmoothed datasets obtained from the previously developed instrument, and this resulted in a rank ordering of the algorithms according to performance on SNR, peak width and resolution over a range of smoothing window sizes. It was decided to use the three best performing algorithms out of the set of six for further study.

The third body of work is an extension of the second and it involves deriving a novel method for using 2 algorithms of different mathematical structure and different optimal smoothing window sizes to smooth datasets with very high noise, very low SNR and very fast DAQs of the order 3 kHz and above. These latter datasets used for testing the methods were obtained from independent sources and contained completely hidden and unknown signals. The method uses these pairs of algorithms simultaneously and not successively, and significant improvements in performance on SNR, peak width and resolution when compared to the best-performing algorithm in each pair were obtained.

The simultaneous-algorithm smoothing method has shown some demonstrable advantages in its ability to extract greater signal information from high-density, low SNR fast DAQ datasets by *post-facto* data processing using open source software. Finally, the potential application of such programmable mathematical techniques to the further development of more efficient small-scale inexpensive instrumentation is discussed.

TABLE OF CONTENTS

1 Appraisal of Micro-Total Analysis Systems with High-Frequency and Low-Resolution Data Acquisition

1.1 Preface: Cheap Miniaturisation gives Bad Signals

The aim of this preface is to outline the sequence of events and areas of study which formed the basis of this candidature. It is hoped that giving a timeline and explaining the development of thinking during the last 4 years, will lend context and clarify what might otherwise seem to be a puzzling jump from engineering an instrument, to the more esoteric study of the mathematics of signals and its application.

At the beginning of this candidature, the initial threefold intention was to

- build a miniaturised capillary electrophoresis instrument which could be operated independently and remotely. The purpose was to
- further obtain data from such remotely placed in instrument directly and in real time. Further the initial aim was to
- build this instrument from self-made and off-the-shelf components as cheaply as possible.

The original threefold task was partially (perhaps mostly) successful. Certainly the instrument was demonstrated to be small, workable, low cost, and could be independently powered for at least 3 days. During the process of building, testing and benchmarking the instrument it became apparent that a 10 bit ADC was too coarse to obtain useful data on its own, and so the aim changed from just building a cheap working system to building one which could perform better by addressing the SNR issue with a low-cost and low-resolution ADC.

1.1.1 Low-Cost Miniaturisation

The details of the cost of all components and the total cost of the instrument are dealt with in Chapter 2. A survey of the literature cited in Chapters 1 and 2 shows that it is difficult to find any specific reference to the total cost of an inexpensive instrument, as

very few papers seem willing to come up with an overall figure. Many times authors will state that the cost is low, without specifying exactly what it is. This may seem puzzling at first, because it can be well understood by any reader that the cost of an instrument built in 2003 will rise with inflation, but it remains a simple arithmetic matter to project inflation over a period of 15 or 20 years to give an estimate of what the current cost would be to build a similar instrument. Having said this however, in many cases the cost is likely to be less than expected because electronic components have become relatively cheaper as they become more ubiquitous.[1] They also frequently improve in quality, smaller size, reliability and speed; so it is probably understandable that authors and experimenters tend to avoid the issue.

What was more frequently encountered were devices which appeared to have low cost, but were often far bulkier and heavier than they need have been. Frequently such instruments mimicked expensive commercial benchtop apparatus to varying degrees (with improvements in some cases) in their method of operation and rough configuration, reducing costs dramatically by using cheaper components and simplified operating systems.[2] Some illustrations of such devices appear in APPENDIX 1.

Further directions seemed to produce instrumentation which was simplified and could conceivably be made very cheaply whilst also providing convincing results. The factor which seemed to raise its head in such cases was that the instrument was frequently fragile and dependent on careful intervention by the operator in order to obtain the data.[3]

With these difficulties and constraints in mind, as well as facing a steep learning curve during 2014, the instrument described in Chapter 2 was constructed and benchmarked.

1.1.2 Signal Processing

During the application of the instrument constructed in Chapter 2, the small size and compactness of the instrument necessitated a short capillary length (15.0 cm) and an electric field of around 400 $V.cm^{-1}$. Application of an inexpensive (<$40) ADC with only 10 bits of resolution initially resulted in the very coarse electropherograms shown in Figure 1.1.2.1 and Figure 1.1.2.2 which follow. It began with the electropherogram in Figure 1.1.2.1 where the ADC was programmed to read at DAQ = 10.0 Hz to imitate that of a commercial bench-top instrument.

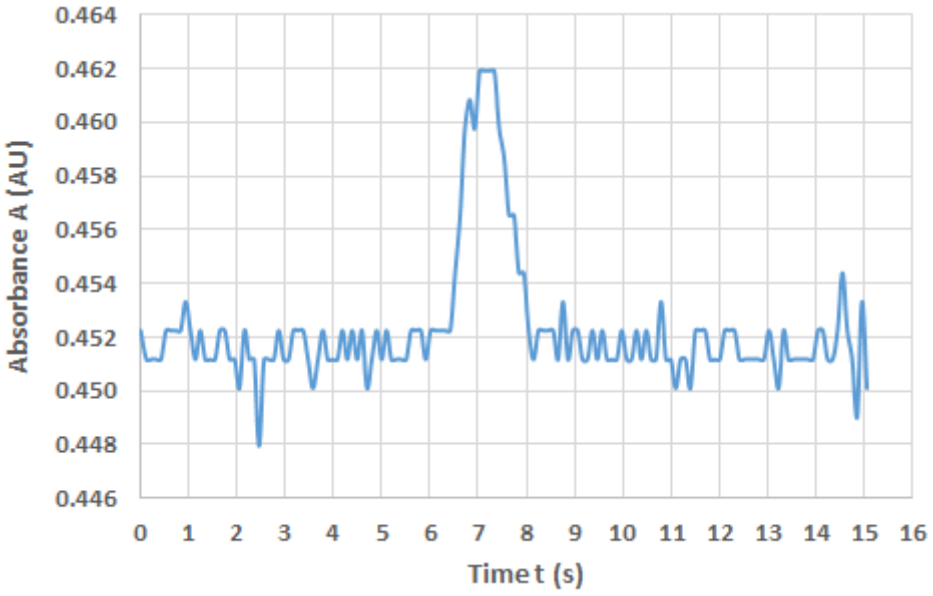


Figure 1.1.2.1: Example of an initial coarse electropherogram using the inexpensive 10 bit ADC programmed to give DAQ = 10.0 Hz without any attempt to address the question of bad resolution. At a migration time of around 7 seconds for the principal signal, visual inspection makes it clear that this low resolution is of little use.

To address the low quality of the signal in Figure 1.1.2.1 above, two things became apparent:

- the DAQ would need to be increased dramatically to get as much information as possible in order to deal with the noise distortion of the principal signal (shown in Figure 1.1.2.2 below);
- increased resolution was needed without the expense of purchasing a higher resolution ADC, or significantly compromising the signal migration times, peak widths, peak heights or areas.

  So the 10-bit Arduino UNO being used was applied under the same conditions as in Figure 1.1.2.1 above, but running with the ADC at maximum speed that the data acquisition program would allow.

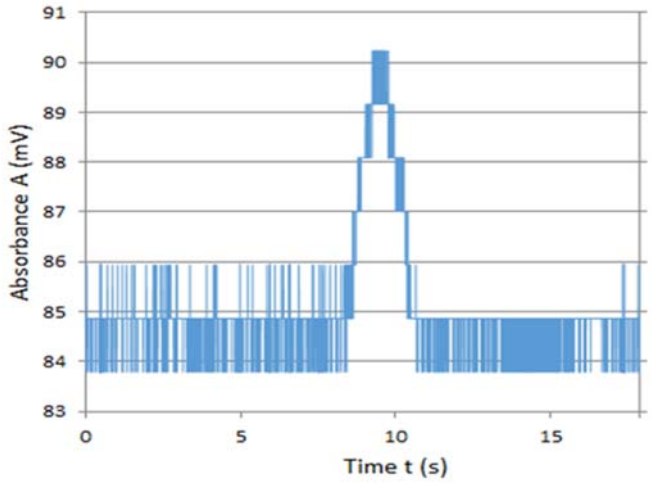


Figure 1.1.2.2: Example of an initial coarse electropherogram again using the same cheap 10 bit ADC, but unprogrammed, and running at maximum speed to give DAQ ≅ 1.1 kHz. At a migration time of around 9.5 seconds for the principal signal, this dataset contains a great deal more information, and was judged to be more useful, provided a means could be found to determine a baseline and eliminate most of the noise.

To get the maximum value from such a high frequency signal, the primary data itself needs to be addressed without intrusion, which comes in two forms:

- Software intrusion such as the insertion of programmed pauses in the primary data acquisition before anything else in order to lower the DAQ, or
- Hardware intrusion by inserting a data-intrusive device to artificially smooth the signal before ADC or direct analog processing. It is a simple matter to insert a data intrusive device such as a secondary chip with a higher bit rate, but such data-intrusive devices will contain their own unknown and non-measurable internal errors.[4]

When SNR is high and noise is low, internal errors due to a smoothing chip or circuit may be insignificant, but when SNR is low and noise is high what is more likely is that determination of low concentration analytes becomes troublesome when smoothing errors are unknown. The only remaining option for treatment of noise is to go back to the raw data and find a method of making sense of any information contained within the noise as far as possible.

1.2 Introduction: Aims, Capacity and Emphasis

The overarching aim of this work is to provide a narrative overview of over 4 years of study. There are three principal areas of study, namely the

a) construction and testing of an independently-powered miniaturised apparatus to conduct capillary electrophoresis using off-the-shelf components and a capillary of shorter length than standard desktop instrumentation; (Chapters 2 and 3) with particular emphasis on the detection of low levels of heavy metals.

b) application of the constructed instrument and other similar devices to the acquisition of datasets at "high" and "very high" DAQ (these terms are defined for the purposes of this study in the *List of Abbreviations and Definitions* above) at low resolution and very low SNR and initial *in situ* and *post-facto* processing of this data using five common smoothing algorithms (Savitsky-Golay, Boxcar, Gauss, Geometric Mean and Exponential Mean) and comparing these to the performance of the Nyquist method (used in general physics, astronomy and acoustics) applied to the same data; (Chapter 3) and

c) mathematical development and application of a system of nested iterative pairs of simultaneous smoothing algorithms taken from the above list. These pairs are applied separately to large amounts of data ($\sim 10^6$ data points) with very low resolution signals, acquired at rates of around 6.0 kHz. What follows is the removal of noise with an attempt at minimal compromising of data integrity, and each nested pair is evaluated in terms of performance on three principal criteria, namely

- signal to noise ratio SNR;
- Peak width Wp; and
- Resolution (all in Chapter 4 and 5)

The study begins with an appropriately selective overview of relevant developments in these three areas up to 2018.

1.3 A General Timeline, and some Significant Markers

Microfluidics deals with systems that process or manipulate small (10^{-9} L to 10^{-18} L) volumes of fluids, by using channels with radii mostly in the region 5 μm to 200 μm.

Before commencing a discussion on the progress of the fields, it may be useful to present a schematic figure which represents some selected advances in both miniaturised capillary electrophoresis, and in signal processing up to 2018 which are relevant to this work. Two recent and very promising developments in miniaturisation arise with a Canadian/American group reported in 2010[5] and another in 2018 by Pan, Fang, et.al.[6] Both of these instruments are pictured in APPENDIX 1.

In Table 1.3.1 which follows, some important dates and outcomes are listed, which are important to this work in particular, and specifically to some of the methods used to construct the instrument in Chapter 2; they also reference relevant directions and developments in 3-D printing for future development of such an instrument as outlined in Chapter 6.

In Table 1.3.2, the separate issue of signal smoothing is addressed separately from miniaturised instrumental development. Signal smoothing and noise removal is not unique to chemistry, but is an issue embraced by audionics, astronomy, physics, statistics, engineering and related fields.

Table 1.3.1: Some important dates and selected developments in the fields of miniaturisation, component integration, and fabrication. The events outlined below are cited in detail in References in Chpters 1 - 5 of the text.

	Miniaturisation, Component Integration, Fabrication, and 3D Printing
Before 1991	Micromachining, Early attempts at portability by scaling down benchtop instruments and breaking them into components for dismantling and transport for on-site analysis.
1991 to 2000	1992: Manz et al; Planar chips technology for miniaturization and integration—capillary electrophoresis on a chip. 1993: Emanuel Sachs - patents the first 3D printer Expansion of multilayer soft lithography fabrication 1995: Caliper Technologies Corp. - formed by Manz, Bock et. al. - Commercialisation of LOC technology and development.
2001 to 2010	2001: Nachamkin et.al. - Use of newly-released Agilent Bioanalyzer 2100 for restriction fragment length polymorphism analysis of a gene. 2001: Schlautmann et.al; Powder-blasting for microfabrication of CE chips with integrated conductivity sensors. 2001-2003: Development of relief moulds for valves and pumps were refined using standard photolithography. 2003: Wheeler, et al; Microfluidic device for single-cell analysis. 2003: Sia & Whitesides -Microfluidic devices fabricated in PDMS. PDMS is compatibile with biological materials. 2003: Jackson et al. develop a miniaturized CE instrument with amperometric detection; integration of glass CE chip, battery, dual-source HV power,, interface circuit and modules, with a total size of ca. 100 × 150 × 25 mm. 2004: After acquisitions and growth, Caliper Tecgnologies becomes Caliper Life Sciences. Concentrates on using its patents to provide new products for microfluidics in biotechnology and pharmacology. 2006: Sandia Laboratories markets an early portable CE-based system. Improvements in valve and micromixing technology 2008: Manz et.al; μTAS - Further developments in cost, size and efficiency
2011 to 2018	2011: Caliper Life Sciences acquires NovaScreen Biosciences (2005), Xenogen (2006), Cambridge Research & Instrumentation (2011). CLS is then acquired by Perkin-Elmer. 2016: Mahdi et.al. Highly sensitive flexible SnS thin film broad spectrum (UV-IR)photodetector. 2017: Expansion of frequency-specific LEDs 2016: Most efficient miniaturisation claimed by Canadian/US group. See APPENDIX 1. 2018: Total size bioanalyzer was reduced to 90 × 75 × 77 mm; cost ≅ $500, See APPENDIX 1.

Development of portable and miniaturised CE and related microfluidic devices almost seem to have been growing in two directions, where portability and miniaturisation are historically often seen as separate issues. Miniaturisation does not necessarily mean portability, and this developmental tension can be seen at various times shown in Table 1.3.1. The rise and fall of Caliper Technologies from 1995 until the acquisition of its last incarnation as Caliper Life Sciences by Perkin-Elmer in 2011 is perhaps illustrative of such difficulties.

Table 1.3.2: Some important dates and selected developments in the fields of DAQ and Signal Processing. The events outlined below are cited in detail in References in Chapters 1 - 5 of the text.

Smoothing, Algorithms and DAQ	
Before 1991	1964: Savitsky and Golay write their seminal paper. 1990: Gorry - Applies general least-squares smoothing and differentiation by the Savitzky-Golay method to applicable chemical signals. 1990: Archibald et.al. - Application of Holt-Winters method to signal smoothing. 1990: Shinagawa, et.al. – Important early analysis of *Jitter* in high-speed Sampling Systems
1991 to 2000	1991: O'Haver – Introduces a systematic method of using a PC for signal processing in chemical measurement. 1996: Kitagawa et.al. - Monte Carlo filter and smoother introduced for non-Gaussian signals. 1997: Hu, Gosine et.al. - The first eigenstructure method for pattern recognition and sinusoidal signal retrieval in white noise. 1999: de Levie, et.al. – First use of Excel's Solver for estimating parameter precision in non-linear least squares. (S-G)
2001 to 2010	2004: Halmer, von Basum, et.al. - Fast EMS algorithm for real-time instrumental use 2005: Bernabé-Zafón, Torres-Lapasió, et.al. - Two-way background correction by cubic smoothing and multivariate data analysis in CE. 2006: Du, Kibbe et.al. use wavelet transform-based pattern matching to identify unpatterned noise in order to filter it out. 2007: Solis, Campiglia et.al. – First use of multiple-pass moving average algorithm for baseline estimation in CE and application to baseline correction on real-time data. 2008: Dasgupta, et.al. - Peak resolution using Microsoft Excel Solver. Extension of the work of de Levie (1999)
2011 to 2018	2012: Laude et.al. – Data filtering method in the analog domain, using a quasi-Nyquist method. 2016: Salmasi, Buttner et.al. Novel method of fractal analysis for low SNR.signals. 2017: Armstrong et.al. Propose a method of total peak shape analysis as a basis for signal enhancement and minimising influence of noise.

1.4 Miniaturised Capillary Electrophoresis: Construction, Limitations and Improvements

Attempts to miniaturise capillary electrophoresis have met with varying success and have been a recurring theme for over 25 years.[7]

Two of the most troublesome limitations of capillary electrophoresis when used to detect low concentrations (low intensity signals) are:

a) poor sensitivity[8], and

b) data smoothing which does not have statistically significant impact on such low-intensity signals.

Smoothing is said to be *data-intrusive* when a smoothing device is inserted between the detector and the data output. Such a device may be hardware such as RC circuits, or chip-embedded software. Either way, the experimenter has no control over the smoothing process, and receives data which is non-primary and has already been filtered in some way. Non-intrusive smoothing occurs when the only device between detector and experimenter is the ADC. Digitised data is evaluated, and the experimenter is then enabled to decide on the best *post-facto* smoothing algorithms. The criteria which can be used to make such a decision are discussed in chapters 3 and 4.

1.4.1 Construction

Recent advances in three-dimensional printing have allowed the construction of miniaturised capillary electrophoresis to progress far beyond innovative micro-machining[9] or subsequent powder-blasting[10] approaches (used independently of or as an adjunct to) micro-machining. Although the fabrication of the outer casing and some of the internal sections can be done using low resolution 3D printing, the high-resolution fabrication of microchannels of the order 10 μm to 100 μm at room temperature remains a challenge.[11]

1.4.2 Limitations

What has become interesting (and hence popular) since about 2001 is the advent and increasing frequency of 3-D printing as a method of trialling different designs and configurations in miniaturised instruments. Initially this new technology was expensive and slow, but papers since 2001 and referred to in Table 1.3.2 above have shown an increasing interest in this technology as cost has reduced and flexibility (such as simultaneous printing using different materials and different degrees of resolution at the same time) has increased. The reporting on 3D printing of PDMS by Hinton, Hudson et.al.[12] shows that such a process can still be highly complex at the present time. It does however demonstrate the workability of 3-D printing in such a useful material but what these and other authors make apparent, is that before an entire miniaturised device can be printed in a single process, what will be needed is the development of new materials which have properties at least as useful as those of PDMS whilst also being amenable to the rigours of much higher resolution extrusion printing.

The portable CE-based system of Sandia Laboratories (2007) demonstrated proof of concept and was one of the first, but it did not include additional end-on functionality such as genetic amplification, which when integrated effectively with CE, provides the basis for useful additional medical diagnostics beyond the most frequent call for blood chemistry profiles.[13]

Miniaturised portable instrumentation until about 2012 then concentrates on demonstration of principles, applications for specific purposes, and an ongoing battle with sensitivity, achieved by development of both detectors and LEDs of various configurations. During the reading of some recent literature on this topic, it seems clear that variance contributions from detector electronics may impact on the peak height, peak width and symmetry,[14] and this appears to be true even when light in the transducer is highly coherent.[15]

What also becomes clear is that there are difficulties in packing HV circuits in close proximity to ADC and fluid control systems together in a small package. Perusal of the paper by Renzi[16] shows that there is a great deal of unexpected complexity in the design, engineering, and manufacture of such a package. It is not surprising therefore that robustness and independent operation as essential criteria are often left by experimenters for later development.

What also becomes apparent is that independence from human intervention in the collection and processing of data is a difficult technical issue.[17]

In miniaturised electrophoresis, the problem of injection method was identified as a very difficult one. The methods considered were pneumatic injection, syringe-powered injection, and direct peristaltic pump injection. Of the three, the latter was selected after trial of the others, whose disadvantages included:

- the complexity involved in re-filling syringes, and the addition of cross-flow switching valves.[18-19]
- the requirement for control sequences[20] in the software. Pneumatic injection also requires one or more valves and gas interface gaskets.[21] It was also rejected because of additional weight, mechanical complexity and additional complexity in software control.

1.4.3 Improvements

Improvements relevant to this study may be categorised into three main divisions:

a) Design.

This includes robustness, efficiency of operation, minimum number of operating components and micro-machines.

Sandia Laboratories produced a portable CE-based system as early as 2005,[16] and this system was commercially available from 2006, but operation meant the regular presence of a human operator. The work by Lu and Collins[22] demonstrates the feasibility of metal-ion determination using a microchip fabricated from borosilicate glass, whose advantages are claimed to include improvements in speed, cost, portability, automation and solvent/sample consumption. Some ideas from this work were incorporated in the design and implementation of the instrument described in Chapter 2, and are referenced in that chapter.

More recently, instrumentation of Sáiz et al.[23] and that shown by Van Schepdael[17] go some way to achieving design improvements by reducing component size, power consumption, weight and therefore increasing portability.

Photodiodes[24] with fast response times[25-26] together with similar advances in frequency-specific[27] LED technology means that reliability, power consumption[28] and miniaturisation have all improved.[29]

b) Cost.

The principal area of cost reduction lies in the use of inexpensive, off-the-shelf components which are both low-cost and easily deployed.[30] Interestingly, it seems as if the ubiquitous Moore's Law may be at work in the field of CE and microfluidics as ADC and HV chips become smaller, faster and cheaper. Off-the-shelf components are also increasing in reliability and robustness, whilst decreasing in both size and cost.[31]

c) Size.

Reduction in instrument size can be achieved with both short capillary length (≅15 cm) and more efficient integration of chips and control circuits. One problem with size reduction is the proximity of HV application across the capillary without sparking, undesired earthing or unwanted circuits developing through BGE and electrolyte channels.[32] This is a very difficult issue. As far as the other components are concerned, fortunately there are increasing levels of miniaturisation of useful technology.[33]

d) Performance

Improved detectors with significantly increased sensitivity remain an active area of research and development.[34] A high linear dynamic range achieved with very sensitive detectors, combined with very low baseline noise (<50 μAU) achieved with physical stability, instrument casing which acts as a Faraday Cage, and high intensity LEDs which may allow for detection of impurities as low as 0.05% of a main peak. Use of such newer (post-2016) LEDs combine the advantage of frequency-specific choice with a removal of the problem of lamp lifetime and lamp fragility which often plagues standard bench-top commercial instrument analysis systems.[5]

In the instrument constructed in Chapter 2, an electric field strength found to be most useful at about 400 $V.cm^{-1}$ was chosen compared to variable (often substantially higher) values in benchtop usage, in order to allow for high sensitivity,[35] short migration times, and to minimise the chances of sparking noted above.

1.5 Low SNR signals in high-noise background: Identifying the Issues

In the Preface of §1.1.2 above, reference was made to the poor quality of signals obtained from a 10 bit ADC running at fast DAQ. It was quickly realised that to detect low amplitude signals within such a rough high noise background would be a challenge which needed to be met by a comprehensive study of both low amplitude signals and high background noise in a way which would give maximum value to an experimenter.

1.5.1 High Noise

After initial signals were received from early hydrodynamic injections on the miniaturised instrument, at first it seemed as if the digitisation of noise shown in Figure 1.1.2.2 above was an insurmountable problem because noise was converted into one of two states as shown on either side of the signal. Some thought and exploratory mathematics was then undertaken after reading some of the literature cited in this chapter. Examination of the low resolution noise showed that it was not uniform but contained regions of higher or lower density as can be seen by visual inspection of Figure 1.1.2.2. It was then decided to treat this low resolution noise as clustered information which could be explored using statistical techniques. Fortunately, some background in the digitisation of classical musical instruments pointed in the direction of the Nyquist theorem. Nyquist's result is that data with two or more points per cycle in high DAQ, allows increasing fidelity of reconstruction with increasing DAQ via the Cardinal Theorem of Interpolation.[36]

The problem of removing noise in such an environment led to the exploration of the idea that rather than treating noise in an isolated way as irrelevant rubbish which needed to be removed, it could be removed more sensitively by increasing the resolution using the Nyquist method outlined in Chapter 3. An alternative way could be to use the methods of Du, Kibbe, et. al. and use wavelet transform-based pattern matching[37] to identify and then filter out unpatterned noise.

1.5.2 Low Signal

The next question which arose during acquisition of electropherogram data from the miniaturised instrument was to find a method for low limits of detection and accuracy.[38] To analyse peak shape and size meaningfully, a set of points on either side of the peak is required[39] in order to measure SNR. After the work of Chapter 2 was completed, attention was focused exclusively on analysis of signals and noise. The 5 commonest algorithms were tested on low resolution signals and then compared in performance with the Nyquist algorithm, and this process is outlined in Chapter 3. This allowed a rank ordering of these six algorithms in terms of performance. A method was then developed which used nesting of algorithm pairs taken from the three best-performing of these six algorithms. These nested pairs were firstly applied to some early low resolution data sets obtained during the work of Chapter 2. Initial results were promising, and then these data sets were applied to very high DAQ datasets with very high noise so that all signals were completely obscured from any visual inspection. What was realised in Chapters 3, 4 and 5 was that noise is not simply a standard deviation along the y-axis but at high DAQ, slight fluctuations along the time axis also occur; such noise is known as *jitter*.[40] Whilst jitter may be reduced by modulating the time variability (X- direction), such an approach was rejected because it would be data intrusive, and if data intrusion of conventional Y direction noise was rejected, then for consistency it also needed to be rejected in the X direction.

The performance of these pairs of nested algorithms applied to such anonymous datasets is assessed and documented in Chapter 5.

1.6 Conclusions

During this introductory chapter, the following main themes were considered within the framework of a narrative of 4 years' work.

1. An overview of developments, difficulties, limitations and progress in miniaturised Capillary Electrophoresis, particularly in its application to the detection of heavy metal contaminants in exposed environments.
2. Description of the building and application of a miniaturised capillary electrophoresis instrument with simple design at low cost, and using cheap off-the-shelf components.
3. Outlining the difficulties encountered in the interpretation and application of low resolution and high noise signals from the instrument.
4. Addressing these difficulties in signal elucidation with the application of a hitherto underutilised algorithm, and comparing and evaluating its performance to 5 smoothing algorithms commonly used in capillary electrophoresis.
5. Development of a method for using 2 simultaneous algorithms on high noise and low SNR electropherograms in a previously untested way.

1.7 References

1. Whitesides, G. M., The origins and the future of microfluidics. *Nature* **2006,** *442* (7101), 368-373.

2. Marini, R. D., Rozet, E., Montes, M. L., Rohrbasser, C., Roht, S., Rheme, D., Bonnabry, P., Schappler, J., Veuthey, J. L., Hubert, P., Rudaz, S., Reliable low-cost capillary electrophoresis device for drug quality control and counterfeit medicines. *J Pharm Biomed Anal* **2010,** *53* (5), 1278-1287.

3. Chiang, M. T., Lu, M. C., Whang, C. W., A simple and low-cost electrochemiluminescence detector for capillary electrophoresis. *Electrophoresis* **2003,** *24* (17), 3033-3039.

4. Grinias, J. P., Whitfield, J. T., Guetschow, E. D., Kennedy, R. T., An Inexpensive, Open-Source USB Arduino Data Acquisition Device for Chemical Instrumentation. *J Chem Educ* **2016,** *93* (7), 1316-1319.

5. Kaigala, G. V., Bercovici, M., Behnam, M., Elliott, D., Santiago, J. G., Backhouse, C. J., Miniaturized system for isotachophoresis assays. *Lab on a chip* **2010,** *10* (17), 2242-2250.

6. Pan, J. Z., Fang, P., Fang, X. X., Hu, T. T., Fang, J.,Fang, Q., A Low-Cost Palmtop High-Speed Capillary Electrophoresis Bioanalyzer with Laser Induced Fluorescence Detection. *Sci Rep* **2018,** *8* (1), 1791-1802.

7. Harrison, D. J., Fluri, K., Seiler, K., Fan, Z., Effenhauser, C. S., Manz, A., Micromachining a Miniaturized Capillary Electrophoresis-Based Chemical Analysis System on a Chip. *Science* **1993,** *261* (5123), 895-897.

8. Breadmore, M. C.; Tubaon, R. M.; Shallan, A. I.; Phung, S.-C.; Abdul Keyon, A. S.; Gstoettenmayr, D.; Prapatpong, P.; Alhusban, A. A.; Ranjbar, L.; See, H. H.; Dawod, M.; Quirino, J. P., Recent advances in enhancing the sensitivity of electrophoresis and electrochromatography in capillaries and microchips (2012–2014). *Electrophoresis* **2015,** *36* (1), 36-61.

9. Tay, F. E. H., van Kan, J.A., Watt, F., Choong, W.O., A novel micro-machining method for the fabrication of thick-film SU-8 embedded micro-channels. *J. Micromech. Microeng.* **2001,** *11*, 27-32.

10. Schlautmann, S., Wensink, H., Schasfoort, R., Elwenspoek, M., van den Berg, A., Powder-blasting technology as an alternative tool for microfabrication of capillary electrophoresis chips with integrated conductivity sensors. *J. Micromech. Microeng.* **2001,** *11*, 386-389.

11. Parekh, D. P., Ladd, C., Panich, L., Moussa, K., Dickey, M. D., 3D printing of liquid metals as fugitive inks for fabrication of 3D microfluidic channels. *Lab on a chip* **2016,** *16* (10), 1812-1820.

12. Hinton, T. J., Hudson, A., Pusch, K., Lee, A., Feinberg, A. W., 3D Printing PDMS Elastomer in a Hydrophilic Support Bath via Freeform Reversible Embedding. *ACS Biomater Sci Eng* **2016,** *2* (10), 1781-1786.

13. Kaigala, G. V., Hoang, V.N., Stickel, A., Lauzon, J., Manage, D., Pilarski, L.M., Backhouse, C.J., An inexpensive and portable microchip-based platform for integrated RT–PCR and capillary electrophoresis. *Analyst* **2008,** *133* (3), 331-338.

14. Castro, E. R., Manz, A., Present state of microchip electrophoresis: state of the art and routine applications. *J Chromatogr A* **2015,** *1382*, 66-85.

15. Kakkar, A., Rodrigo Navarro, J., Schatz, R., Pang, X., Ozolins, O., Udalcovs, A., Louchet, H., Popov, S., Jacobsen, G., Laser Frequency Noise in Coherent Optical Systems: Spectral Regimes and Impairments. *Sci Rep* **2017,** *7* (844), 1-10.

16. Renzi, R. F., Stamps, J., Horn, B.A., Ferko, S., Van der Noot, V.A., West, J.A.A., Crocker, R., Wiedenman, B., Yee, D., Fruetel, J.A., Hand-Held Microanalytical Instrument for Chip-Based Electrophoretic Separations of Proteins. *Anal. Chem.* **2005,** *77* (3), 435-441.

17. Van Schepdael, A., Recent Advances in Portable Analytical Electromigration Devices. *Separations* **2016,** *3* (2), 1-12.

18. Huynh, B. H., Fogarty, B. A., Nandi, P., Lunte, S. M., A microchip electrophoresis device with on-line microdialysis sampling and on-chip sample derivatization by naphthalene 2,3-dicarboxaldehyde/2-mercaptoethanol for amino acid and peptide analysis. *J Pharm Biomed Anal* **2006,** *42* (5), 529-34.

19. Debets, A. J. J., Mazereeuw, M., Voogt, W. H., van Iperen, D. J., Lingeman,H., Hupe, K.-P., Th. Brinkman, U. A. , Switching valve with internal micro precolumn for on-line sample enrichment in CZE.pdf. *J Chromatogr* **1992,** *608*, 151-158.

20. Kiplagat, I. K., Kuban, P., Pelcova, P., Kuban, V., Portable, lightweight, low power, ion chromatographic system with open tubular capillary columns. *J Chromatogr A* **2010,** *1217* (31), 5116-5123.

21. Kelly, R. T., Wang, C., Rausch, S. J., Lee, C.S., Tang, K., Pneumatic microvalve-based hydrodynamic sample injection for high-throughput, quantitative zone electrophoresis in capillaries. *Anal. Chem.* **2014,** *86*, 6723−6729.

22. Lu, Q., Collins, G.E., Microchip separations of transition metal ions via LED absorbance detection of their PAR complexes. *The Analyst* **2001,** *126* (4), 429-432.

23. Saiz, J., Duc, M. T., Koenka, I. J., Martin-Alberca, C., Hauser, P. C., Garcia-Ruiz, C., Concurrent determination of anions and cations in consumer fireworks with a portable dual-capillary electrophoresis system. *J Chromatogr A* **2014,** *1372C*, 245-252.

24. Mahdi, M. S., Ibrahim, K., Hmood, A., Ahmed, N.M., Azzez, S.A., Mustafa, F.I., A highly sensitive flexible SnS thin film photodetector in the ultraviolet to near infrared prepared by chemical bath deposition. *RSC Advances* **2016,** *6* (116), 114980-114988.

25. Tsujikawa, T., Funamoto, H., Kataoka, J., Fujita, T., Nishiyama, T., Kurei, Y., Sato, K., Yamamura, K., Nakamura, S., Performance of the latest MPPCs with reduced dark counts and improved photon detection efficiency. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* **2014,** *765*, 247-251.

26. Pandey, K., Chauhan, M., Bhatt, V., Tripathi, B., Yadav, P., Kumar, M., High-performance self-powered perovskite photodetector with a rapid photoconductive response. *RSC Advances* **2016,** *6* (107), 105076-105080.

27. Belusic, G., Ilic, M., Meglic, A., Pirih, P., A fast multispectral light synthesiser based on LEDs and a diffraction grating. *Sci Rep* **2016,** *6*, 32012-32020.

28. Pan, Q., Hu, H., Zou, Y., Chen, M., Wu, L., Yang, D., Yuan, X., Fan, J., Sun, B., Zhang, Q., Microwave-assisted synthesis of high-quality "all-inorganic" CsPbX3 (X = Cl, Br, I) perovskite nanocrystals and their application in light emitting diodes. *Journal of Materials Chemistry C* **2017,** *5* (42), 10947-10954.

29. Zhang, X., Zhang, J., Wu, X., Lv, Y., Hou, X., Light-Emitting-Diode-induced Chemiluminescence Detection for Capillary Electrophoresis. *Electrophoresis* **2009,** *30* (11), 1937-1942.

30. Duong, H. A., Le, M. D., Mai-Nguyen, K. D., Hauser, P. C., Pham, H. V., Mai, T. D., In-house-made capillary electrophoresis instruments coupled with contactless conductivity detection as a simple and inexpensive solution for water analysis: a case study in Vietnam. *Environ Sci Process Impacts* **2015,** *17* (11), 1941-1951.

31. Minhass, W. H., Pop, P., Madsen, J., Blaga, F.S. In *Architectural Synthesis of Flow-Based Microfluidic Large-Scale Integration Biochips*, Proceedings of the 2012 International Conference on Compilers, Architectures and Synthesis for Embedded Systems - . , Lyngby, Denmark, DTU Informatics

Technical University of Denmark: Lyngby, Denmark, 2012; pp 181-191.

32. Urban, P. L., Universal electronics for miniature and automated chemical assays. *Analyst* **2015,** *140* (4), 963-975.

33. Ryvolová, M., Preisler, J., Brabazon, D.,; Macka, M., Portable capillary-based (non-chip) capillary electrophoresis. *TrAC Trends in Analytical Chemistry* **2010,** *29* (4), 339-353.

34. Wu, T.-L., Meen, T-H., Chao, S-M., Ji, L-W., Shih, L-C., Huang, C-H.. Tsai, J-K., Wu, T-C., Application of ZnO micro rods on the composite photo-electrode of dye sensitized solar cells. *Microsystem Technologies* **2017,** *24* (1), 285-289.

35. Wahl, J. H., Goodlett, D.R., Udseth, H.R., Smith, R.D., Use of small-diameter capillaries for increasing peptide and protein detection sensitivity in capillary electrophoresis-mass spectrometry. *Electrophoresis* **1993,** *14* (4), 448-457.

36. Candes, E. J., Romberg, J., Tao, T., Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on Information Theory* **2006,** *52* (2), 489-509.

37. Du, P., Kibbe, W. A., Lin, S. M., Improved peak detection in mass spectrum by incorporating continuous wavelet transform-based pattern matching. *Bioinformatics* **2006,** *22* (17), 2059-2065.

38. Shackman, J. G., Watson, C.J., Kennedy, R.T., High-throughput automated post-processing of separation data. *Journal of Chromatography A* **2004,** *1040* (2), 273-282.

39. Wahab, M. F., Patel, D. C., Armstrong, D.W., Peak Shapes and their Measurements - The Need and the Concept behind Total Peak Shape Analysis. *LC•GC Europe* **2017,** (December 2017), 670-678.

40. Shinagawa, M., Akazawa, Y., Wakimoto, T., Jitter Analysis of High-speed Sampling Systems. *IEEE Journal of Solid-State Circuits* **1990,** *25* (1), 220 - 224.

2 Low-Cost Miniaturised Electrophoresis for Continuous Environmental Monitoring

2.1 Introduction

In the literature, reference is frequently made to social changes which favour increasing levels of miniaturisation of useful technology,[1] and there is potential in chemical analysis systems which are cheap and easily deployed.[2] Reliability and predictive capacity of environmental modelling is highly dependent upon data from multiple locations with a high temporal resolution. Cheap and portable instrumentation (ideally, instrumentation which is autonomously and/or remotely operated) is essential to be able to provide the temporal resolution required for the development of accurate and relevant models.

Miniaturisation of analytical instrumentation as an ideal has been a recurring theme for over 20 years,[3] and there has been success in the development of small-scale (i.e. significantly smaller than a standard bench-top commercial instrument) analysis systems.[4] The majority of useful examples to date are around single-purpose analytical sensors (pH for example) that can detect a single species. The need now is to further extend on some of this ground-breaking work to produce instrumentation which is able to reliably detect and measure multiple species, but the advantages of such apparatus must lie in more than just miniaturisation. For example, the development of the Miniaturised Total Analysis System (μ-TAS) by Manz et al.[5] has influenced systems for genetic analysis, clinical diagnostics, chemical synthesis, environmental monitoring and drug screening. More recently, the approach of Tavares da Costa et al.[6] clearly demonstrates the feasibility of using an off-the-shelf programmable microprocessor to run a miniaturised analysis system. They demonstrate short analysis times which then require only small sample volumes for analysis, and short migration times mean that potentially large datasets can be obtained from continuous running of apparatus.

2.1.1 Possible Suitability of μCE

Plume and Outfall Modelling is an area of interest, because:

- It requires large amounts of continuous data from field collection in order to predict the mixing and broader interaction of pollutants discharged from a point source within their receiving environment, particularly in riverine, marine, estuarine, and mining regions.
- The transitory nature of such phenomena means that there is a need for cheap, sophisticated miniaturised instrumentation to monitor changes in real time.
- Real-time or near-real-time environmental monitoring provides information about human impact on plants and animals,[7] or on marine, estuarine and riverine systems, and an opportunity to mitigate further damage.

All the above are currently done mostly by collecting samples from places of interest at discrete points in time. If sampling is not done at a correct place and time, then potentially damaging events may not be detected.[8] Continuous monitoring enables profiling of changes within such environments, and the acquired data is useful for the establishment of control systems, remediation, amelioration or future prevention. Representative examples include: Acidic Mine Drainage (AMD)[9], nutrient and fertiliser run-off,[10] mining outfall,[11] and industrial and manufacturing waste.[12]

To summarise; in the mining industry as well as in wider marine, estuarine and riverine environments, there is a need for low-cost instrumentation which is flexible in both application and location, and which can be operated remotely. Since it can be used for both cationic and anionic detection using a single instrument applied to the same sample, Capillary Electrophoresis (CE) has the potential to fulfil this and similar needs. For example the work by Lu and Collins[13] demonstrates the feasibility of metal-ion determination using a microchip fabricated from borosilicate glass, where they demonstrate advantages which include improvements in speed, cost, portability, automation and solvent/sample consumption. A simple photomultiplier tube (PMT) is applied to read absorbance. Latterly, the emergence of more efficient, smaller, and much cheaper photodiodes[14] with fast response times[15-16] together with similar advances in frequency-specific[17] LED technology[18] means that reliability, power consumption and miniaturisation have all improved.[19]

Although miniaturisation of CE to varying degrees and in various configurations has been achieved, autonomous or remote operation together with independence from

human intervention in the collection and processing of data is a harder technical issue.[20]

Compactness of instrumentation presents no particular advantage when instruments are confined to a laboratory, or when portability requires disproportionate ancillary support in the form of data collection hardware, human on-site operation and cumbersome power requirements. Whenever larger portable instrumentation and ancillary support are used, or samples are collected and processed in a laboratory, the primary issue of independence and concurrent issues of convenience and cost are not met adequately.

2.1.2 Preview and Outline of Instrumental Requirements

To arrive at a design which fulfilled all the requirements listed above, different commercial and bespoke designs from commercial and academic literature (cited below) were considered. What quickly becomes apparent is that the complexity, weight, and size of the instrument are not principally determined by electrical considerations such as provision of high voltage or absorbance measurements by voltage output - both of which can be easily miniaturised and controlled by microchips and efficient circuitry - but by the mode of injection, sample and buffer positions, and power supply via battery pack or other independent means.

Miniaturisation could be more useful in applications similar to those above if it were enhanced by satisfying the following (or similar) requirements:

- *Independently Powered and Lightweight* – which also means that the apparatus should have low ancillary requirements in terms of human operation and intervention. Earlier portable instruments such as the CEP-5100 (CE-P2) Autosampler (from EH Systems Llc10031 Pelham Road, Simpsonville SC 29681, USA) and instrumentation such as that of Sáiz et al.[21] have and those shown by Van Schepdael[20] go some way to achieving such an outcome.
- *Robustness* – which means that the engineering of the instrument needs to be such that it can operate automatically and reliably over a wide range of temperatures and in potentially harsh outdoor conditions. Repair and replacement of components needs to be infrequent, simple and cheap;

- *Accessibility* – which means that both software commands to the apparatus, and data uploading from the apparatus should be simple, open-source and able to be reliably and remotely controlled;
- *Cost* – reduction in size of the apparatus is only beneficial in this category if there is a proportional reduction in cost, which implies cheap electronics, power and mechanical components;
- *Accuracy and Precision* – which means that miniaturisation, portability, and accessibility cannot be allowed to lead to significant reduction in operating quality;
- *Sensitivity* – which means that low levels of concentration from the order ppb can be reliably detected and identified.

2.1.3 Initial Hardware Engineering of Instrumental Prototype

To go as far as possible to address the requirements outlined above, a first instrumental prototype was constructed by using cheap off-the-shelf components and an in-house designed operating system using the following instrumental configuration:

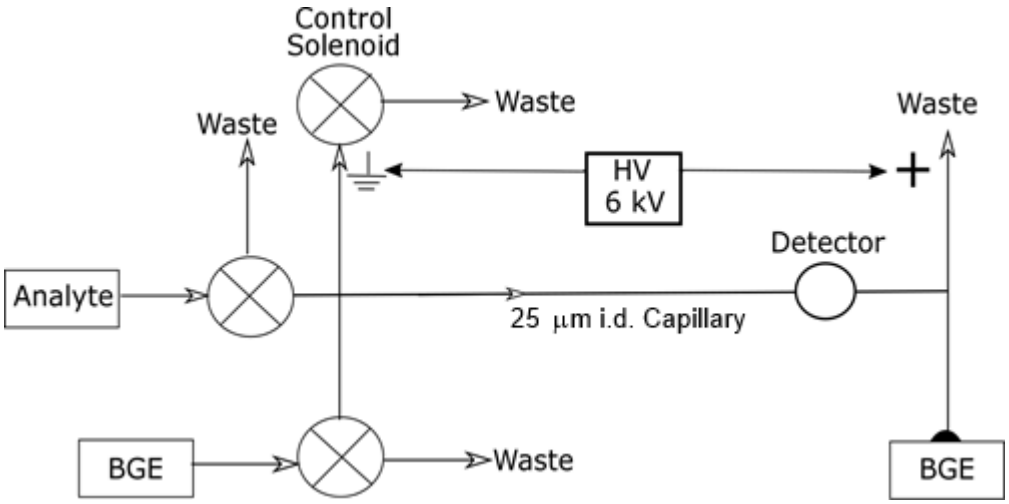


Figure 2.1.3.1: Schematic of the initial prototype of the apparatus showing the relative positioning and interconnection of all components. A more detailed schematic of the apparatus using photographs of the major components, and a photograph of the apparatus can be seen at Figure Apx 2.2 and Figure Apx 2.3 respectively in APPENDIX 2.

Electrophoresis was performed in a fused silica capillary with $L_T = 15.0$ cm; $L_D = 12.5$ cm, 25 μm i.d. and 363μm o.d. with standard polyimide coating (Cat. No. TSP 025150, Polymicro Technologies™ Phoenix, AZ). This small diameter and short length together

with an electric field strength of 400 V.cm^{-1} compared to common benchtop usage was chosen in order to allow for high sensitivity[22] and shorter migration times.

One 4-way and one 3-way Interconnect Cross CapTite™ connectors (LabSmith, Inc., 6111 Southfront Road, Suite E, Livermore, CA 94551) were modified by drilling the BGE and injection channels to a diameter of 1.0mm, leaving the 360µm o.d. capillary connection unchanged. The capillary and other analyte and buffer flow connections were made using PEEK™ One-Piece Fittings for connecting capillary or tubing to CapTite™ components as shown below:

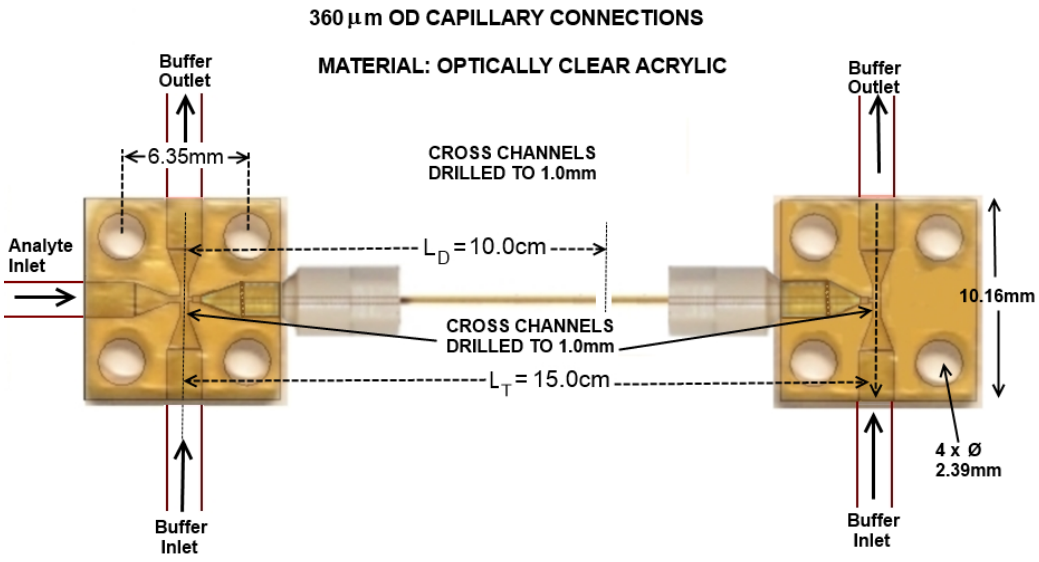


Figure 2.3.3.2: Interconnect Cross CapTite™ Connectors, showing modifications with buffer and analyte inlets and outlets.

Colorimetric detection was accomplished by using a partially purpose-built detection module as follows: A blue LED light source (483nm, Luckylight Electronics part number: LL-504BC2E-B4-2CC, LED-TECH.DE optoelectronics GmbH Schürmannshütt 38 C, D-47441 Moers, Germany) was oriented directly above and as close as possible to the 2mm capillary window. This was achieved by using a commercial Agilent interface (part number: G7100-60400, Agilent Technologies, 5301 Stevens Creek Blvd. Santa Clara, CA 95051 United States); light passing through the capillary window was collected using a microscope objective lens and directed on to a miniature Integrated Photodiode Detector (Part number: IPL 10530 DAL; Photomatrix Limited, Paceycombe Way, Poundbury, Dorchester, Dorset, DT13SY, UK) which was connected to a custom-built absorbance detector and amplifier. Absorbance measurements in the electropherograms are given in terms of the photodiode output voltage (mV), and are

proportional to the absorbance of the blue LED light through the capillary separation channel.

The HV supply consisted of two compact ULTRAVOLT (UltraVolt, Inc, 1800 Ocean Avenue, Ronkonkoma, NY 11779) high voltage power supplies connected in series, (3V5-N0.5-EI-W and 3V5-P0.5-EI-W) These two modules were connected via a common earth, and 5.0V input, giving a potential of 0 to -3000V and 0 to +3000V, respectively, providing a total potential difference of 6.0kV. The system was modified to be independently powered by a 12.0V nickel/cadmium battery, normally used to power a portable electric drill, with a capacity of 1.5Ah (Akku BCC 1212 battery, Hitachi Koku, Tokyo, Japan) fitted with a voltage regulator for production of a constant 12 V output to power the detector, together with a resistor array for the tapping off 5.0V (used to power the HV power supply) and 4.0V (to power the solenoids and peristaltic pumps). The system was run with the battery during separation of Coumerin-102, fluorescein and FITC described in §2.3.2 and shown in Figure 2.3.2.1 below. The pH values of all solutions were measured using a labCHEM v.1.02 pH meter. (TPS Pty Ltd. Jamberoo Street, Springwood, Brisbane, Australia, 4127)

2.1.4 Software Engineering of Instrumental Prototype

To minimise cost and increase simplicity, non-proprietary software was used to control the instrument through the Arduino board. The Arduino microchip programming language (based on C++) was used to facilitate component control via the operating system.[6] Data was also read with the Arduino board, collected and saved as a .csv file using open-source software named “Processing”. The Operating System was written in a program structured with instrumental initialisation, followed by 12 distinct subroutines - each controlling a single sequential step in the electrophoretic process.

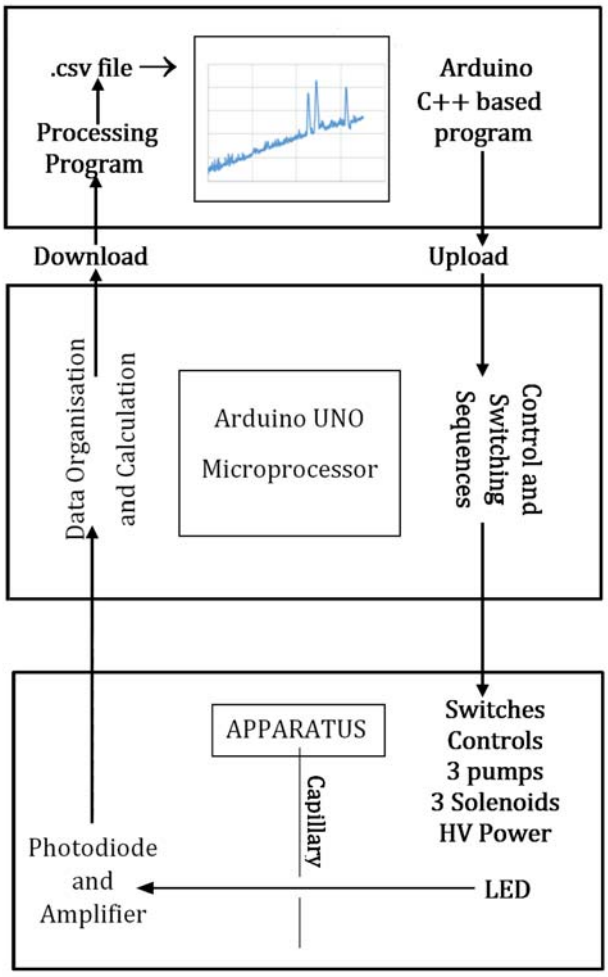


Figure 2.1.4.1: Schematic diagram of the apparatus software control and operation. The details of the coding in each step can be seen in APPENDIX 2.

The stepwise, sequential operation of the instrument is first outlined under the heading "OPERATIONAL OUTLINE" in APPENDIX 2. This is followed by the complete coding of the Operating System. As each step is executed, it is logged on the screen output, enabling the operator to monitor the progress of each run. Finally, data is written in a .csv file in a spreadsheet in real time during the electrophoresis. This spreadsheet can then be pre-programmed to show the electropherogram. This notion of a pre-programmed spreadsheet will be explored further in Chapter 5.

2.2 Prototype Modification and Standardisation

The first question to ask was: "Does it work?" and with that in mind, it was decided to test the apparatus by initially running dyes such as fluorescein and rhodamine to test the following, and standardise against common commercial instrumentation:

- Resolution
- Injection
- Sensitivity

2.2.1 Testing Detector Performance and Improving Resolution

The instrumental prototype was standardised against the Agilent 35900E (Agilent Technologies, 5301 Stevens Creek Blvd. Santa Clara, CA 95051 United States) commercial system. The Agilent 35900E system runs using a RS-232 microchip with a 24-bit analog/digital converter and the Agilent Interface in model 35900E has maximum baud rate of 38400 baud. The baud rate is the number of signal changes that occur per second, which in this case is the number of times per second that the Arduino or Agilent board samples the voltage from the photodiode measuring the absorbance.

The Arduino has 10-bit resolution, but a maximum baud rate of 115200 baud. An upscaling of baud rate – even at the low resolution of 10 bits increases resolution. To test this, two Arduino boards were connected in parallel with an Agilent 35900E ADC and the same hydrodynamic injection of fluorescein/tetraborate.

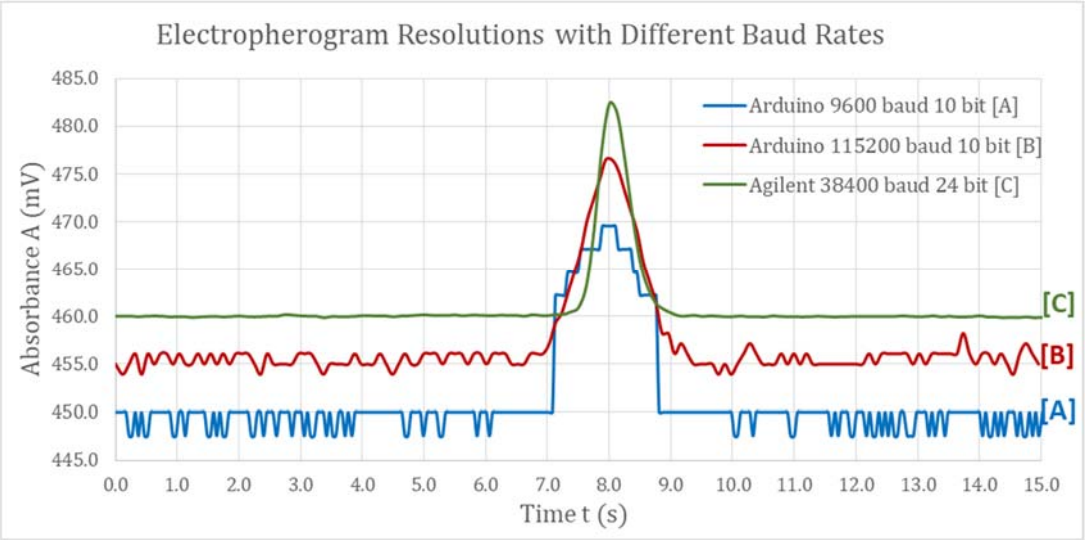


Figure 2.2.1.1: Comparative resolutions of the same simultaneous electropherogram using two Arduino baud rates and the standard Agilent 35900E ADC.
SNR[A] = 18.0 SNR[B] = 28.4 SNR[C] = 432.9

An intuitive examination of these results would seem to indicate that bits of resolution has a far greater influence on SNR than an increase in baud rate.

To increase the bits of resolution further on a 10-bit Arduino board, there seem to be two ways of doing this:

1. By the addition of a hardware "hack", which is a circuit added in parallel to the Arduino UNO in order to increase the number of bits of resolution. Such a circuit from a hackers' website is shown in APPENDIX 2, Figure Apx 2.1. This method was rejected for reasons explained at the figure in APPENDIX 2, and also in the references to data-intrusive sampling elsewhere in this text.

2. By an embedded software method, using an application of the Nyquist theorem, as explained below.

It was decided to apply the Nyquist Theorem. The computational method used to achieve this oversampling uses an adaptation of the following formula, which converts the analog input (converted to a 10-bit signal), and then processes this to give an output voltage scaled to the 1.1 V (1100 mV) internal reference of the Arduino board:

$$A = \frac{1100}{2^{10+n}} \left(\frac{\sum_{j=1}^{4^n} d_j + 2^{n-1}}{2^n} \right) \qquad (2.2.1.1)$$

Where A is proportional absorbance (in mV);

n is the number of *additional* bits (i.e. number of bits more than 10) of resolution;

j is the number of data points which need to be processed by the software to achieve the desired increase in resolution;

d_j are individual data readings from the photodiode assembly.

The rationale and mathematical justification for this approach is discussed further in depth and application in Chapter 3 and Chapter 5 respectively.

To test this approach to resolution, and simultaneously to evaluate the performance of the detector and the ADC capabilities of the Arduino, a comparison was devised as follows:

- FIA was performed using the prototype instrument in a series of consecutive hydrodynamic injections of 1.0 mM Rhodamine-A in 10.0 mM phosphate buffer (adjusted to pH ≅ 5.0), using the phosphate buffer as the carrier.
- Data was collected from the in-house constructed LED absorbance detector on the prototype instrument.
- This data from the LED absorbance detector was processed by the ADC from a standard commercial instrument; Agilent 35900E at 24 bits ADC and 38400 baud, with data sampling at 10.0 Hz.
- The prototype instrument processed the data using the Arduino ADC at 14.84 bits ADC and 115200 baud, with data sampling also at 10.0 Hz. This use of fractional bits allows increased resolution by oversampling using the Arduino ADC, with data processed down to 10.0 Hz using a new algorithm that will be discussed in full in Chapter 3.

Data collection using the two ADCs was simultaneous, with three successive detection traces (from a total of 10 traces) from the two ADCs shown in below. The figure shows detector responses, with the signal-to-noise ratio for all peaks being on average 1.06 times higher for the 24-bit Agilent ADC compared to the oversampling 14.84-bit Arduino.

Since the injections were controlled by the Arduino software, they are shown to be at precisely the same intervals. The more serious question is illustrated in the difference between peak heights and peak areas; this shows inconsistency in injection volume which was a major area that needed to be addressed.

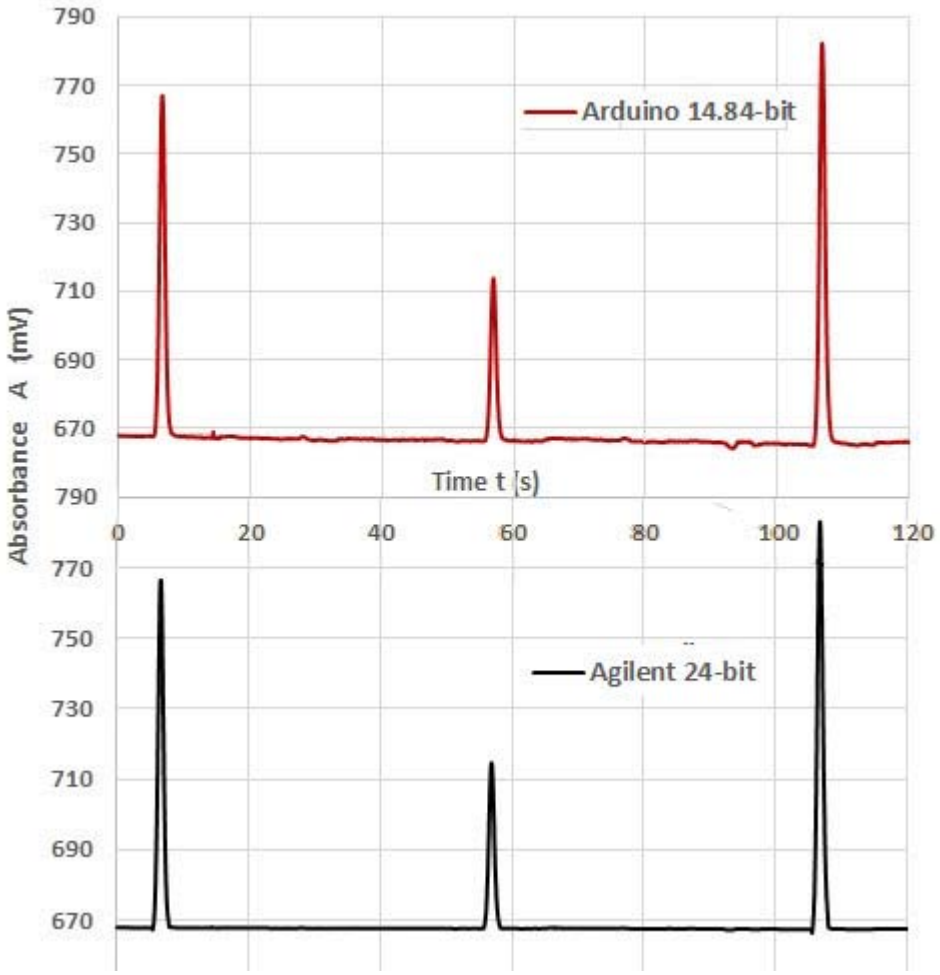


Figure 2.2.1.2: Comparison of migration times and resolution of commercial (Agilent) system using ChemView, and Arduino-controlled instrument using the self-written operating system. Flow injection analysis: Capillary i.d. = 25µm; L_D=20 cm; L_T=30 cm; Analyte 1.0 mM Rhodamine-A; Phosphate buffer 10.0 mM; pH = 5.0.
For the 10 traces from which the above three traces are extracted, the ratio of SNR Agilent trace/SNR Arduino trace = 1.062±0.015

2.2.2 Testing and Improving Injection

Figure 2.2.1.2 also illustrates another major issue with the instrument that needed to be overcome – the varying peak height. The RSD for peak areas of the 10 peaks used to standardise the prototype was 34%. The source of this unacceptable variation is the single-cam rotation in the pump, which results in a cyclic variation in pressure.

Injection was not happening at the same point in the cam rotation, and so injection pressure was not reproducible. Several approaches were considered to rectify this:

- A commercial "lab-bench" - type injection using injection syringes. This was rejected because of high levels of energy consumption involving the use of geared electric motor drivers (reducing battery life), the additional software complexity involved in re-filling syringes, and the addition of switching valves.[23-24]
- Pressurised gas-driven injection. Injection using gas pressure entails use of a small gas cylinder in order to achieve portability. A gas cylinder then involves a pressure regulator which includes a pressure sensor and a servo motor in order to adjust pressure, all of which require control sequences[25] in the software. It was also rejected because of additional weight, mechanical and software complexity.
- Electrokinetic injection was attempted, and gave far better reproducibility[26] of peak area and peak height than single-cam pump injection. This approach was discounted because of known bias and susceptibility to variations in sample matrix composition. It also resulted in a shortening of battery life.
- Introduction of a pulse dampener. A self-constructed pulse dampener at first appeared to solve the problem completely.

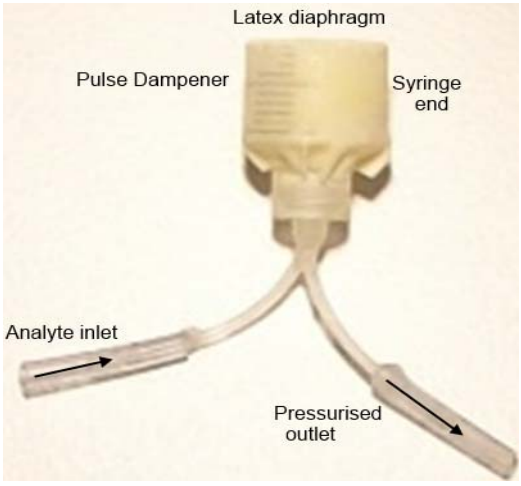


Figure 2.2.2.1: The self-constructed (negligible cost) pulse dampener using part of a cut-off 15 mL syringe and the finger of a latex glove. Insertion of this pulse dampener involved addition of one extra solenoid switching valve and only two lines of extra software code. The pulse dampener was pressurised for 50 seconds before injection.

Initially, reproducibility and stability of peak areas and peak heights were excellent, because the short injection time compared to the large volume of the pulse dampener resulted in a decrease in RSD from 34% to 9.5%. However, after 60 runs, there was a decrease in peak height and this was due to weakening of the diaphragm after repeated use. This approach was not pursued further because of this and because of the high cost of commercial pulse dampeners.

- Finally, an approach was implemented that exploits the very characteristic of the peristaltic pump which caused the problem in the first place. With the rotation of the single-cam there is a variation in the power input depending upon where it is in the cycle. This cyclic variation in power consumption (and so also a measure of the cam position in the pump cycle) can be monitored by measuring the voltage consumption of the pump during operation and thus, it can be used to time the injection such that it occurs in the same position in the pump cycle.

Figure 2.2.2.2 shows the variation in the pump voltage during operation, which as indicated above, is not uniform. It is however periodic[27] and there is a regular spike in the voltage at a particular point in the pump cycle. Using this spike, the pump voltage was monitored and a trigger value was set to ensure that the injection solenoid would be closed at precisely the same position in the pump cycle.

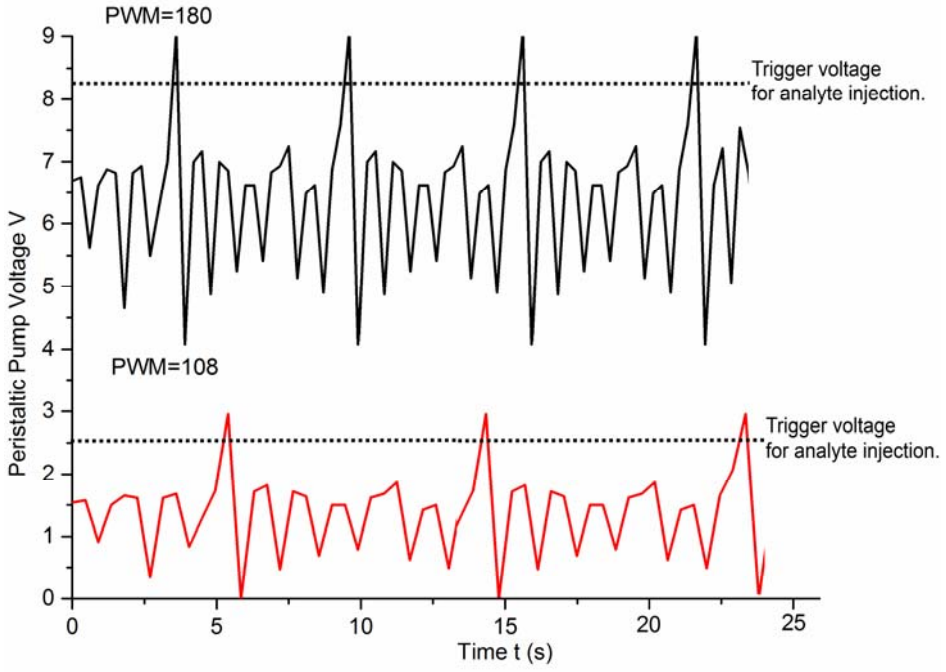


Figure 2.2.2.2: Cyclic variation in voltage (as a measure of power consumption) for single-cam peristaltic pump. When the voltage input is decreased via the Pulse-Width Modulator (PWM) function, the period and amplitude are seen to change accordingly.

From 12 replicate injections, it was shown that peak time, width and height all deliver a marked improvement in RSD when compared to the unregulated injections, with for example, the RSD in peak area decreasing from 49% to 2.9% in table 2.2.1 below.

Table 2.2.2.1: Data set (n=12) for electropherograms of 1.0 mM fluorescein (F) (injection time = 1.8 s) with and without using digital feedback loop to trigger the analyte injection.

	Unregulated Injection			Regulated Injection with Feedback loop		
	Migration time (s)	Peak width (s)	Peak height (mV)	Migration time (s)	Peak width (s)	Peak height (mV)
Average	116.140	1.418	1.477	120.03	1.501	1.489
Std Dev	0.752	0.371	0.726	0.289	0.0113	0.0438
RSD	0.648	26.146	49.136	0.241	0.756	2.943

2.3 Performance of the miniaturised CE

To fully evaluate the performance of the constructed CE, it was used to compare electrophoretic separations of Coumerin-102 (to indicate EOF), fluorescein (F) and fluorescein isothiocyanate (FITC) in both a commercial Agilent CE and the home-built instrument. The sample contained 25 ppb of each of the dyes in 5 mM sodium tetraborate buffer, and was separated in 5 mM sodium tetraborate.

2.3.1 Comparison of Separations using Agilent and Arduino-based Instrument

As can be seen from Figure 2.3.1.1, the separations appear very similar with regard to migration times, resolution and sensitivity. As discussed earlier, the commercial instrument is more sensitive; the concept of sensitivity being a comparative metric, and measured by the ratio:

$$\frac{SNR(Agilent)}{SNR(Arduino)} = 1.52 \pm 0.06$$

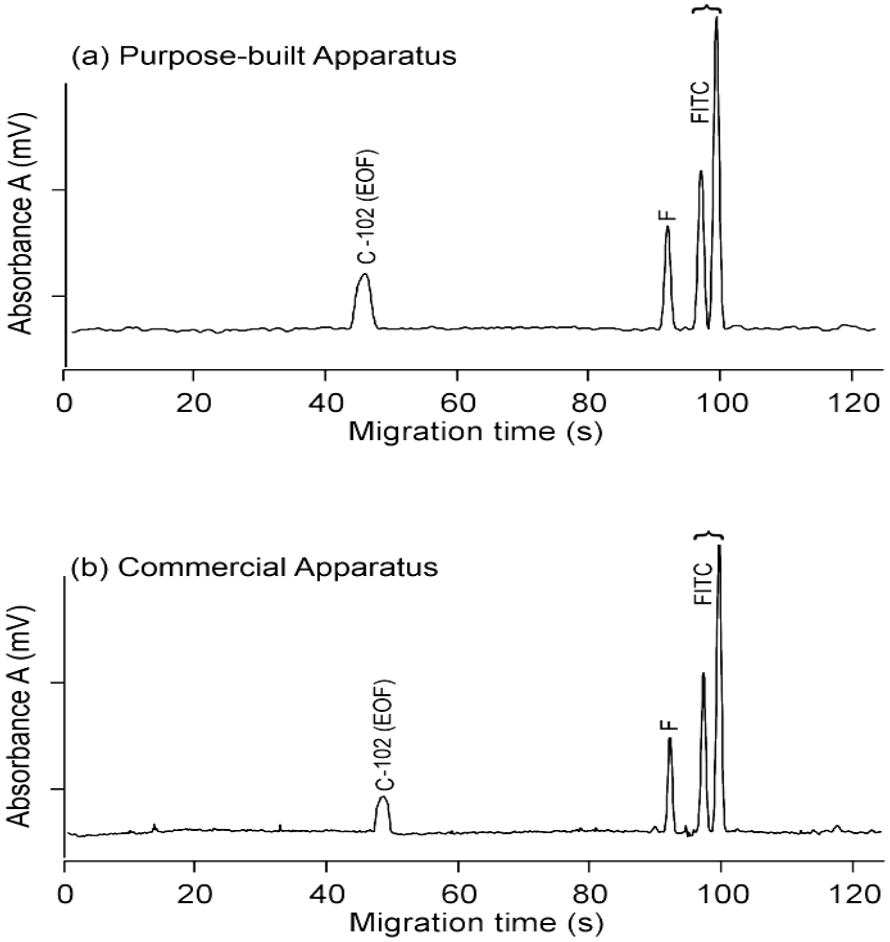


Figure 2.3.1.1: Comparative electropherograms showing separation of fluorescein (F) and fluorescein isothiocyanate (FITC) at analyte concentration of 0.01 mmol/L with 5 mmol.L^{-1} sodium tetraborate buffer pH=9.2. Both systems operated at a sampling rate of 10.0 Hz. For the Agilent system, L_D =12.5 cm; L_T = 32.5 cm; V = 13 kV; E=400.0V.cm^{-1}. For the Arduino-controlled system, L_D = 12.5 cm; L_T = 15.0 cm; V= 6.0 kV; E=400.0V.cm^{-1}.
For the three peaks shown: $\frac{SNR(Agilent)}{SNR(Arduino)}$ =1.52±0.06.

2.3.2 Application to heavy metal detection in drinking water

One of the areas of local significance was selected for the potential application of the miniaturised CE is in the detection of heavy metals in drinking water. North-Eastern Tasmania has reported high lead levels in drinking water that exceed the Australian drinking water requirements of 10 µg/L, with levels as high as 540 µg/L reported by Harvey et. al.[28] Lead can be easily detected by CE after complexation with 4-(pyridylazo) resorcinol (PAR);[29] so a BGE consisting of 10.0 mM sodium tetraborate was selected, based on the stability of the Pb-PAR complex at pH = 9.2.[30]

In order to be directly applicable to water samples on-line, it is necessary to first mix the sample with PAR. This was achieved by using a T-junction to draw both PAR and

sample (~16 times excess PAR) through the analyte peristaltic pump, with the pump also helping to mix the two solutions. This pre-capillary derivatisation took place in an 8.0 cm (i.d. = 2.5 mm) polymer tube after the pump which provided sufficient time for the Pb^{2+} to be fully complexed with the PAR, in a manner analogous to that described by Glatz.[31]

Figure 2.3.2.1 below shows the analysis of tap water from Hobart spiked with 40 µg/L of Pb^{2+} over 3 days of continuous uninterrupted operation. As can be seen from the figure, the migration time repeatability is excellent as is the repeatability of the peak area.

Analysis of the data shows migration time =92.6 s (RSD = 1.76%);

Peak height = 2.8 mV (RSD = 7.6%) and peak area = 6.7 mV.s (RSD = 8.8%)

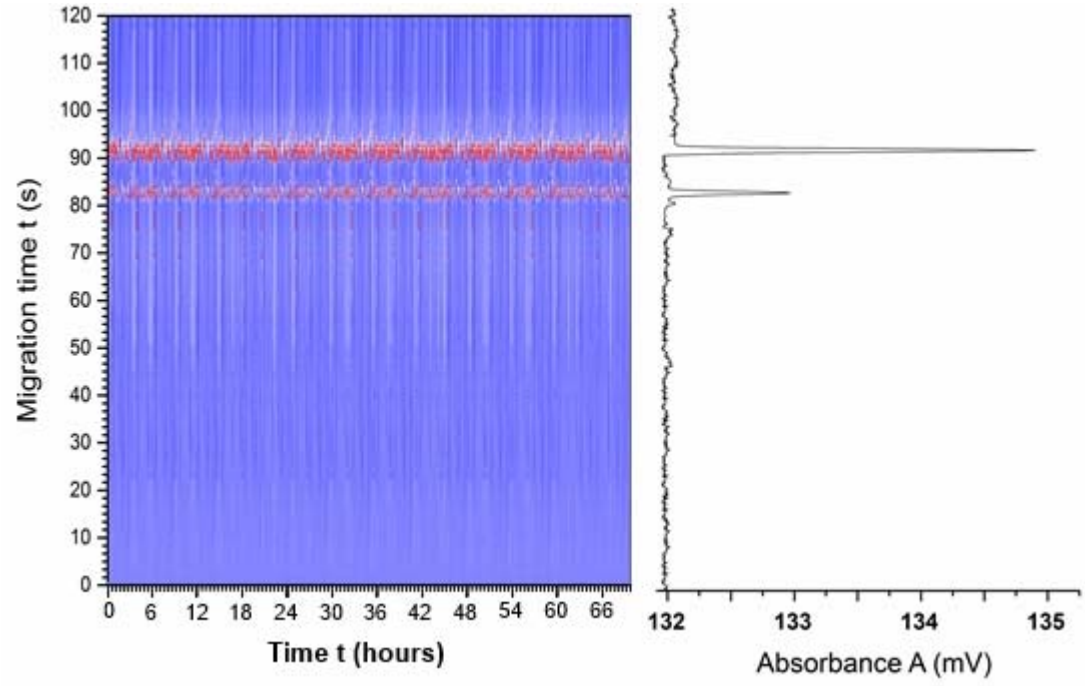


Figure 2.3.2.1: Graphic illustrating repeatability of data in 1163 runs over 3 days. Separation of Lead/PAR in tap water spiked with lead at 40.0 ppb. L_D = 12.5 cm; L_T = 15.0 cm; V = 6.0 kV; E = 400 V/cm; Capillary i.d = 25 µm. BGE is 10.0 mM at pH = 9.2

Analytical performance data of the system can be found in Table 2.3.2.1, which provides a LOD of 3 µg/L, which is sufficiently below the Australian drinking water requirements to be useful for monitoring Pb^{2+} in drinking water.

Table 2.3.2.1: Linear range, regression data, LOD and reproducibility test of Pb-PAR analyte using an internal standard of Cr-PAR at 40.0 ppb. *Note: Reproducibility is calculated based on peak heights at 3 concentration levels: 5μg/L, 50μg/L and150μg/L

Linear range (μg/L)	Correlation coefficient	LOD (μg/L)	LOQ (μg/L)	Repeatability, RSD (%, n=5 at each concentration)					
				Intraday			Interday (5 days)		
				5μg/L	50μg/L	150μg/L	5μg/L	50μg/L	150μg/L
5-150	0.9980	3	9	2.0	3.2	3.8	6.0	7.0	8.2

*Note: Reproducibility is calculated based on peak heights at 3 concentration levels: 5μg/L, 50μg/L and150μg/L

To validate the potential of the self-built CE, drinking water samples were collected from the towns of Scottsdale, Ringarooma, Pioneer and Gladstone in north-eastern Tasmania (from two different locations in each town). After treatment with nitric acid, they were refrigerated and transported back to the laboratory for subsequent analysis. ICP-MS analysis was performed on each sample to validate the CE results, with the data for the 10 samples shown in Table 2.3.2.2.

Table 2.3.2.2: Quantitative comparative results for Pb-PAR in water samples from North-East Tasmania

Sample ID	*Detection by own instrument (μg/L)	Detection by ICP-MS (μg/L)
P1	9.07 ± 0.14	9.55 ± 0.22
P2	26.49 ± 0.98	27.52 ± 0.25
S1	**ND	0.54
S2	**ND	0.30
R1	**ND	0.11
R2	**ND	0.26
G1	**ND	1.23
G2	**ND	0.67

*n=10 at each location, analysed using an Internal Standard of Cr-PAR at 40.0ppb
** Significantly below Australian mandatory drinking water levels of 10 g/L.
Not analysed multiple times.

Lead was detected in only two of the water samples, both from Pioneer, and there is excellent agreement between the ICP-MS and CE results. Electropherograms from these two samples and another from Gladstone with levels below the LOD are shown in Figure 2.3.2.2.

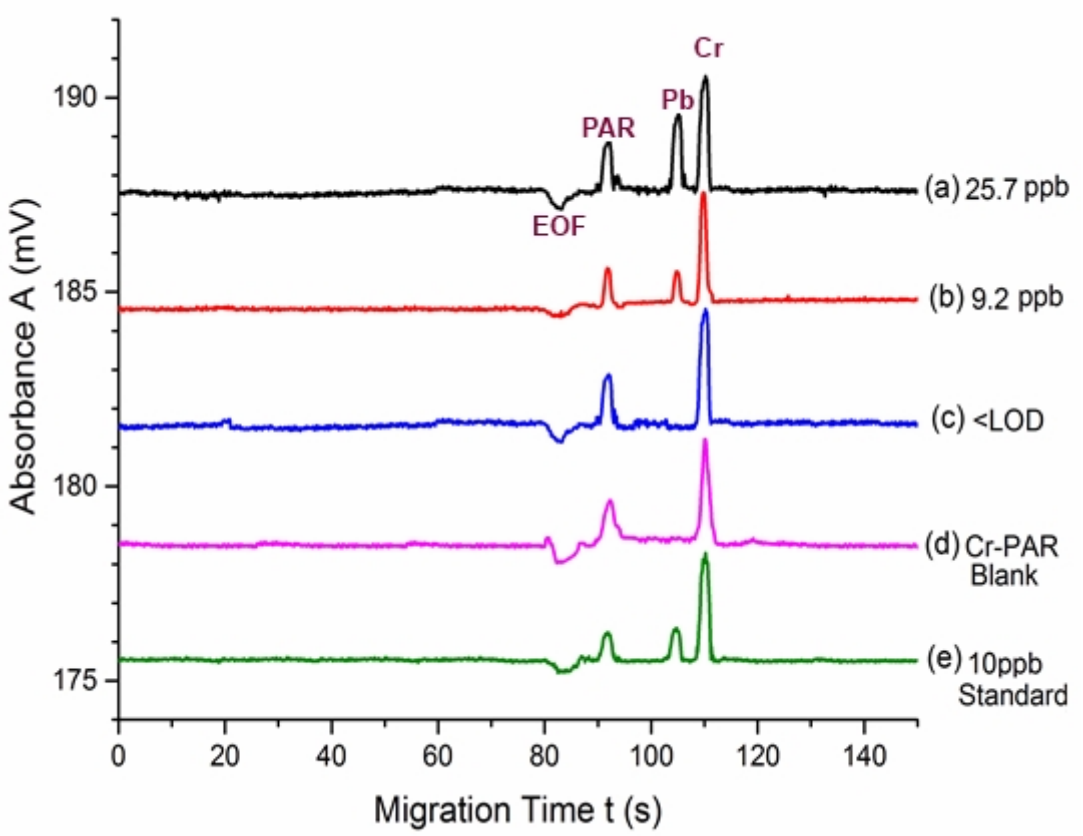


Figure 2.3.2.2: Comparative electropherograms from the purpose-built instrument to detect lead levels in drinking water by measuring absorbance of Pb-PAR showing:
(a) Lead at 27.5 ppb from drinking water at Pioneer;
(b) Lead at 9.2 ppb from drinking water at Pioneer;
(c) Undetectable level below LOD from Gladstone;
(d) Blank Cr-PAR at 40.0 ppb;
(e) Standard 10ppb Australian limit for Pb levels in drinking water.
All electropherograms with $L_D = 12.5$ cm; $L_T = 15.0$ cm; V = 6.0 kV; E = 400 V/cm; Capillary i.d = 25 μm. BGE is 10.0 mM sodium tetraborate at pH = 9.2

2.4 Conclusions

This chapter shows how a compact sequential injection – capillary electrophoresis (SI-CE) instrument prototype was constructed using inexpensive off-the-shelf components (using a 482 nm blue LED, together with cheap off-the shelf isocratic nanoflow peristaltic pumps and flow-switching solenoids) with open source electronic components, and controlled by an Arduino UNO board, with a control operating system written to a Programmable Microprocessor (PMP). The operating system was written to perform the mechanical and electronic steps of the electrophoresis, and then modified to reduce baseline noise by inserting mathematical oversampling and smoothing algorithms. The instrument was also able to operate independently from an inexpensive portable drill battery for three days.

Two miniaturised 3 kV high voltage supplies were used to create a 6 kV differential to facilitate separation. The instrument does not require vials or sample collection, but continuously collects the sample directly. Control of hydrodynamic injection was achieved by monitoring the fluctuating voltage of the single-cam analyte pump and triggering at an optimum level, ensuring consistent injection at the same point in the pump cycle each time. Peak area RSDs were less than 3%, compared to 23% without controlled timed injection. Data collection was performed using the 10 bit ADC converter on the PMP, programmed to create the equivalent of a 15 bit ADC.

The instrument was then calibrated against commercial (Agilent) instrumentation by separation of selected fluorophores at low (0.01mM) concentration.

Repeated operation of the instrument for 3 days (n=1163 runs) showed consistent performance, with migration time and area RSDs $< 2.0\%$ and 9.0%, respectively. It was then shown to be able to detect lead levels in the low (<10 ppb) range, by pre-capillary complexation with PAR, and repeatability was shown by running the instrument continuously for 3 days, measuring 40 ppb lead in spiked tap water.

The potential of the instrument for environmental monitoring was demonstrated by the detection of lead levels in tap water samples taken from areas of concern located in North-Eastern Tasmania.

2.5 References

1. Ryvolová, M., Preisler, J., Brabazon, D.,; Macka, M., Portable capillary-based (non-chip) capillary electrophoresis. *TrAC Trends in Analytical Chemistry* **2010,** *29* (4), 339-353.

2. Duong, H. A., Le, M. D., Mai-Nguyen, K. D., Hauser, P. C., Pham, H. V., Mai, T. D., In-house-made capillary electrophoresis instruments coupled with contactless conductivity detection as a simple and inexpensive solution for water analysis: a case study in Vietnam. *Environ Sci Process Impacts* **2015,** *17* (11), 1941-1951.

3. Harrison, D. J., Fluri, K., Seiler, K., Fan, Z., Effenhauser, C. S., Manz, A., Micromachining a Miniaturized Capillary Electrophoresis-Based Chemical Analysis System on a Chip. *Science* **1993,** *261* (5123), 895-897.

4. Kaigala, G. V., Bercovici, M., Behnam, M., Elliott, D., Santiago, J. G., Backhouse, C. J., Miniaturized system for isotachophoresis assays. *Lab on a chip* **2010,** *10* (17), 2242-2250.

5. West, J., Becker, M., Tombrink, S., Manz, A., Micro Total Analysis Systems Latest Achievements. *Anal. Chem* **2008,** *80* (12), 4403–4419.

6. Tavares da Costa, E., Mora, M. F., Willis, P. A., do Lago, C.L., Hong, J., Garcia, C.D., Getting started with open-hardware:Development and control of microfluidic devices. *Electrophoresis* **2014,** *35*, 2370-2377.

7. Chen, Y., Huang, L., Wu, W., Ruan, Y., Wu, Z., Xue, Z., Fu, F., Speciation analysis of lead in marine animals by using capillary electrophoresis coupled online with inductively coupled plasma mass spectrometry. *Electrophoresis* **2014,** *35*, 1346–1352.

8. Harvey, P. J., Taylor, M. P., Handley, H. K., Widespread Environmental Contamination Hazards in Agricultural Soils from the Use of Lead Joints in Above Ground Large-Scale Water Supply Pipelines. *Water, Air, & Soil Pollution* **2015,** *226* (6), 1-9.

9. Wan, M., Yang, Y., Qiu, G., Xu, A., Qian, L., Huang Z., Xia, J., Acidophilic bacterial community reflecting pollution level of sulphide mine - Wan et. al. *J. Cent. South Univ. Technol.* **2009,** *16*, 0223−0229.

10. Corradini, F., Najera, F., Casanova, M., Tapia, Y., Singh, R., do Salazar, O., Effects of maize cultivation on nitrogen and phosphorus loadings to drainage channels in Central Chile. *Environ Monit Assess* **2015,** *187* (697), 1-17.

11. Locher, H., Mount Lyell Remediation: Sediment Transport in the King River. In *Tasmaninan and Commonwealth Government (Supervising Scientist Report)*, 1997; pp 1-50.

12. Saiz, J., Mai, T. D., Hauser, P. C., Garcia-Ruiz, C., Determination of nitrogen mustard degradation products in water samples using a portable capillary electrophoresis instrument. *Electrophoresis* **2013,** *34* (14), 2078-2084.

13. Lu, Q., Collins, G.E., Microchip separations of transition metal ions via LED absorbance detection of their PAR complexes. *The Analyst* **2001,** *126* (4), 429-432.

14. Mahdi, M. S., Ibrahim, K., Hmood, A., Ahmed, N.M., Azzez, S.A., Mustafa, F.I., A highly sensitive flexible SnS thin film photodetector in the ultraviolet to near infrared prepared by chemical bath deposition. *RSC Advances* **2016,** *6* (116), 114980-114988.

15. Tsujikawa, T., Funamoto, H., Kataoka, J., Fujita, T., Nishiyama, T., Kurei, Y., Sato, K., Yamamura, K., Nakamura, S., Performance of the latest MPPCs with reduced dark counts and improved photon detection efficiency. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* **2014,** *765*, 247-251.

16. Pandey, K., Chauhan, M., Bhatt, V., Tripathi, B., Yadav, P., Kumar, M., High-performance self-powered perovskite photodetector with a rapid photoconductive response. *RSC Advances* **2016,** *6* (107), 105076-105080.

17. Belusic, G., Ilic, M., Meglic, A., Pirih, P., A fast multispectral light synthesiser based on LEDs and a diffraction grating. *Sci Rep* **2016,** *6*, 32012-32020.

18. Pan, Q., Hu, H., Zou, Y., Chen, M., Wu, L., Yang, D., Yuan, X., Fan, J., Sun, B., Zhang, Q., Microwave-assisted synthesis of high-quality "all-inorganic" CsPbX3 (X = Cl, Br, I) perovskite nanocrystals and their application in light emitting diodes. *Journal of Materials Chemistry C* **2017,** *5* (42), 10947-10954.

19. Zhang, X., Zhang, J., Wu, X., Lv, Y., Hou, X., Light-Emitting-Diode-induced Chemiluminescence Detection for Capillary Electrophoresis. *Electrophoresis* **2009,** *30* (11), 1937-1942.

20. Van Schepdael, A., Recent Advances in Portable Analytical Electromigration Devices. *Separations* **2016,** *3* (2), 1-12.

21. Saiz, J., Duc, M. T., Koenka, I. J., Martin-Alberca, C., Hauser, P. C., Garcia-Ruiz, C., Concurrent determination of anions and cations in consumer fireworks with a portable dual-capillary electrophoresis system. *J Chromatogr A* **2014,** *1372C*, 245-252.

22. Wahl, J. H., Goodlett, D.R., Udseth, H.R., Smith, R.D., Use of small-diameter capillaries for increasing peptide and protein detection sensitivity in capillary electrophoresis-mass spectrometry. *Electrophoresis* **1993,** *14* (4), 448-457.

23. Huynh, B. H., Fogarty, B. A., Nandi, P., Lunte, S. M., A microchip electrophoresis device with on-line microdialysis sampling and on-chip sample derivatization by naphthalene 2,3-dicarboxaldehyde/2-mercaptoethanol for amino acid and peptide analysis. *J Pharm Biomed Anal* **2006,** *42* (5), 529-34.

24. Debets, A. J. J., Mazereeuw, M., Voogt, W. H., van Iperen, D. J., Lingeman,H., Hupe, K.-P., Th. Brinkman, U. A. , Switching valve with internal micro precolumn for on-line sample enrichment in CZE.pdf. *J Chromatogr* **1992,** *608*, 151-158.

25. Kiplagat, I. K., Kuban, P., Pelcova, P., Kuban, V., Portable, lightweight, low power, ion chromatographic system with open tubular capillary columns. *J Chromatogr A* **2010,** *1217* (31), 5116-5123.

26. Breadmore, M. C., Electrokinetic and hydrodynamic injection - making the right choice for capillary electrophoresis.pdf. *Bioanalysis* **2009,** *1* (5), 889-894.

27. Deng, R., Cheng, Y., Wang, C-H., Experiments and simulation on a pulse dampener system for stabilizing liquid flow. *Chemical Engineering Journal* **2012,** *210*, 136-142.

28. Harvey, P. J., Handley, H. K., Taylor, M. P., Identification of the sources of metal (lead) contamination in drinking waters in north-eastern Tasmania using lead isotopic compositions. *Environ Sci Pollut Res Int* **2015,** *22* (16), 12276-12288.

29. Regan, F. B., Meaney, M.P., Lunte, S.M., Determination of metal ions by capillary electrophoresis using on-column complexation with 4-(2-pyridylazo)resorcinol following trace enrichment by peak stacking *J Chromatogr B* **1994,** *657*, 409-417.

30. Rahman, I. M. M., Furusho, Y., Begum, Z. A., Sato, R., Okumura, H., Honda, H., Hasegawa, H., Determination of lead in solution by solid phase extraction, elution, and spectrophotometric detection using 4-(2-pyridylazo)-resorcinol. *Central European Journal of Chemistry* **2013,** *11* (5), 672-678.

31. Glatz, Z., On-capillary derivatisation as an approach to enhancing sensitivity in capillary electrophoresis. *Electrophoresis* **2015,** *36* (5), 744-763.

3 Application of the Nyquist Theorem to Chromatographic Analysis: A Comparative Study

3.1 Introduction: Necessary conditions for smoothing

Minimizing noise in chemical measurements is critical for low limits of detection and accuracy.[1] Standardised smoothing algorithms in spectrophotometry – based commercial analytical instrumentation are usually inherent in the instrument hardware, and not subject to change by the user, apart from carefully defined menu options in the controlling software. Analysts may choose parameters such as detection wavelength or sampling frequency, but may not be able to alter detector electronics or be able to change electronically-embedded processing algorithms; such users may be unaware that outcomes can be influenced by such embedded systems.

As data points are read from the detector, a certain minimum number n of data points are needed before any smoothing can occur; at its most basic level, $n = 2$ before we can take an average.

For a set of n data points, hardware-embedded smoothing algorithms usually apply one of the following:

(i) a power function,
(ii) statistical function, or
(iii) polynomial function.

For the purpose of this discussion, n must be at least 3, or else any algorithm becomes trivial; any function (i) – (iii) can be applied to $n = 1$ or $n = 2$ with no useful result.[2]

To achieve meaningful smoothing, it is necessary to remember that every data point occurs in an environment consisting of other points on either side. The data environment used in hardware-embedded smoothing algorithms is mostly numerically symmetrical; smoothing at a particular point is performed by using an equal number of points on either side, so that the total number of points in any smoothed cluster is an odd number. To analyse chromatographic peak shape and size meaningfully, a set of points on either side of the peak is required[3] in order to measure noise and so give numerical context to peak metrics such as peak height from a reliable baseline, or SNR. For instance, a peak consisting of 20 points cannot be usefully described unless there are enough data points on either side to enable meaningful assessment of the

background noise in the detector signal. The question of what precisely is meant by "enough" data points, and "meaningful" assessment will be explored later in this chapter and in Chapter 4.

Spreadsheets may be used to determine a best fit curve through data that may follow a trend, yielding a mathematical relation representing the smallest deviation of selected data from the best fit curve. Hitherto spreadsheets have been used in this way[4] mostly for simple datasets, but if data is noisy, we need mathematical options for reducing the noise in order to make a best estimate of precision and accuracy, and subsequent analysis of critical characteristics.

In this chapter, experimentally collected sets of data by hydrodynamic injection and electrophoretic separation from the instrument described in Chapter 2 are used. Smoothing algorithms are developed and then applied *post-facto* to data points in a spreadsheet to evaluate relative performance of these different smoothing algorithms. This enables

- a structure for the planning of each of the smoothing algorithms and its subsequent coding;

- an examination of possible relationships between essential signal characteristics (such as SNR, peak width, peak width at half-height), smoothing algorithm and smoothing window.

3.1.1 Setting the Scene: A Mathematical Framework for Smoothing

In this chapter

- the underlying mathematics is initially illustrated with concrete examples wherever possible before going into the abstract mathematical structures, in order to make it more accessible to chemists or biologists with undergraduate mathematics who may seek a deeper understanding of algorithmic smoothing techniques;
- the most common algorithms are looked at in terms of their structure and useful outcomes and compared with those of the previously unapplied Nyquist method in order to establish some criteria for choosing the most efficient (i.e. maximum smoothing for least amount of data points) smoothing methods.

3.1.2 A Concrete Basic Example

Start with a set of 26 data points, and apply a smoothing algorithm (such as Gaussian, Savitsky-Golay or some other) to a window (cluster) of 11 points (an odd number) from this set, centred on data point 13.

This means that points 1-7 are untouched, points 8-12 are included, point 13 is the central anchor point, points 14-18 are included and points 19-26 are untouched. The 11-point cluster which is then smoothed around point 13 is the set

$\{(t_8, A_8);.....;(t_{13}, A_{13});.....;(t_{18}, A_{18})\}$ on either side of (t_{13}, A_{13}).

i.e.

$\{(t_{13-5}, A_{13-5});.....;(t_{13}, A_{13});.....;(t_{13+5}, A_{13+5})\}$ on either side of (t_{13}, A_{13}).

The next central point to be smoothed by this shifting 11-point cluster is then point 14.

$\{(t_9, A_9);.....;(t_{14}, A_{14});.....;(t_{19}, A_{19})\}$ on either side of (t_{14}, A_{14}) and so on until the central anchor point has to stop at (t_{21}, A_{21}) before the set runs out of data.

More generally, if $n =$ the number of dataset points, j points are contained in half of the window and k is the central point, then the dataset of interest is:

$$\{(t_{k-i}, A_{k-i});..... ;(t_k, A_k); .....;(t_{k+i}, A_{k+i})\}$$

3.1.3 More General Mathematical Extension

In this study, *fast data acquisition* is defined as data acquired at rates greater than 300Hz, and *low resolution* as data which shows SNR ≤ 2.0 on at least one peak of interest.

Recent advances in computing speed and memory have meant that large volumes of data (of the order GB) are able to be processed in a manner which was hitherto considered to be too complex and long. In this section, we work through a framework for a possible technique using some of the ideas of Morawski,[5] but applying classical algebra rather than the explicit calculus used to treat noise in spectrophotometry. All counters (indicated by subscripts) are from the set of positive integers $\mathbb{Z}^+$. So in what follows, this is written

$$i, j, k, n \in \mathbb{Z}^+$$

By extension, now consider a set of n data points (n =26 in the above example) which starts at t = 0.0 s (where n = 1), and is collected successively in the memory of a microcomputer. To smooth a clustered, odd-numbered subset $2i+1$ of signals (i = 5 so $2i + 1 = 11$ above) we must have $i+1 \leq k \leq n - i$ because we cannot smooth a cluster which is bigger than the set of data we have collected, or try to anchor a smoothing cluster outside the range of the number of data points. (In the above example, this means that we cannot have the centre k of the 11-point smoothing cluster closer to the beginning of the data than position number 6, or closer to the end of the data than position number 21.)

This cluster of $2i+1$ signals is centred around a single (anchor point) signal at position k, namely (t_k , A_k) (where k = 13 in the above example). Let the subscript *"ks*" mean "the smoothed point centred on the k^{th} data point (t_k , A_k) ". The resulting smoothed output centred at the k^{th} data point is (t_{ks} , A_{ks}) , where the point (t_{ks} , A_{ks}) is the result of smoothing by algorithmic operator Ξ on the odd-numbered cluster of points:

$\{(t_{k-i} , A_{k-i}); \ldots\ldots ; (t_k , A_k); \ldots\ldots\ldots; (t_{k+i} , A_{k+i})\}$ on either side of (t_k , A_k).

(For the operator symbol Ξ just read "apply Gaussian" or "apply Boxcar" or any similar smoothing operator.)

Since the actual values of t_i and A_i are not necessarily whole numbers but may be any positive or negative real number from the set $\Re$ of real numbers, then let the subscript *"ks*" mean "the smoothed point centred on the k^{th} data point (t_k , A_k)." This becomes:

$$(t_{ks} , A_{ks}) = \Xi \{(t_j , A_j) | j = k-i, k-i+1, k-i+2, \ldots., \boldsymbol{k}, k+1, k+2, \ldots, k+i ; i, j, k \in \mathbb{Z}^+; t_k , A_k \in \Re \} \quad (3.1.3.1)$$

What this says is:

"The smoothed cluster of points centred on (t_k , A_k) is the result of applying algorithmic operator Ξ to (t_k , A_k) together with the i points before it and the i points after, and the outcome is the new point (t_{ks} , A_{ks})"

The smoothing operator Ξ is straightforward; data is processed in a stepwise monotonic manner, so at each successive point (t_k , A_k) we generate a new point (t_{ks} , A_{ks}) that is a function of (t_k , A_k) and its surrounding data points.

3.1.4 Some Limitations of Window-Based Smoothing (WBS)

A limitation of the window cluster smoothing technique is that it cannot be used on the first $n = 1, 2, \rightarrow k - i$; or last $n - i$ terms of the full dataset, without the adding of values created by some other means. This means effectively extrapolating outside the existing data, and the validity of these sections could therefore be questionable and not a direct representation of the data. The result is that a full data electropherogram will show a discontinuity where the smoothing begins and ends, and to avoid this, data collection needs to start at more than i points before any signal. Failing to do this may mean that some small signals may be lost in the noise in such regions.

In the data used for this study, this possible error source is limited by replacing the noisy absorbance values A_i from $n = 1$ *to* $n = i$, with those from $n = i + 1$ to $n = 2i$ at the beginning of the data; also by replacing noisy absorbance values A_i from $n - i$ to n with those from $n - 2i$ to $n-(i+1)$ at the end of the data. These points are needed, not because they contain signals but because they contain noise, and a smoothing algorithm is not able to begin with half a window without introducing discontinuity. If the number of points n in the full electropherogram is within 20% of n on either side of the signal, then unless this technique is applied, the SNR of the smoothed signal can be compromised by the unsmoothed $2i$ points which are the consequence of the size of the smoothing window.

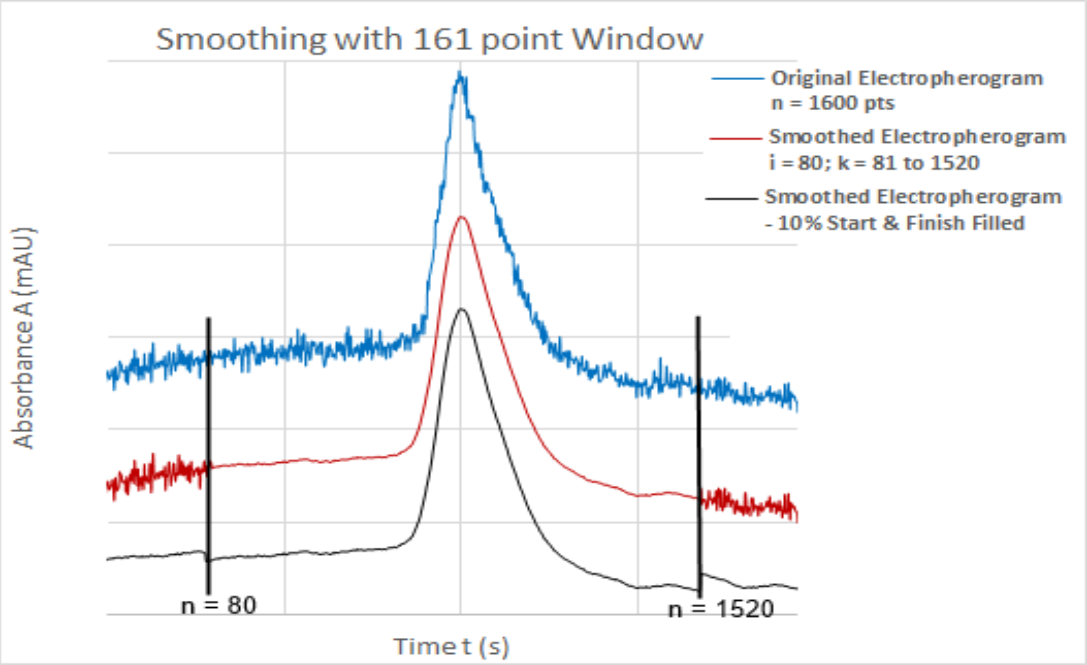


Figure 3.1.4.1: Electropherogram showing smoothing with 161 point window. If small signals with SNR ≅ 3 occur in regions $1 \leq n \leq 80$ or $1520 \leq n \leq 1600$, they will be lost.

With fast data acquisition rate at this low resolution, noise in the original dataset is nearly uniform i.e. the two-dimensional standard deviation σ_{At} at both the beginning 10% of *n* and end 10% of *n* is almost identical; the noise in these beginning and end regions are consistent to within <2%. When the total number of points *n* is such that n>>*2k+1* where *2k+1* is the number of points comprising the region of interest (a satisfactory number of points to determine noise σ_{At}) around a signal, the number of repeated points illustrated in Fig 3.1.4.1 are several orders of magnitude smaller than the cardinal number *n* of the total dataset, and so the artificial completeness illustrated in Fig. 3.1.4.1 above does not interfere with the smoothing of noise on and around the signals. However, it also introduces a phase shift of half the window length into the data; this length is *i*.

The phase shift φ is equal to $\frac{number\ of data\ points}{Data\ Acquistion\ Rate}$; so

$$\varphi = \frac{i}{DAQ} \qquad (3.1.4.1)$$

For example, if the data were all of similar σ_{At} (noise) except for one easily visually identifiable signal, such a peak in the smoothed data would appear half a window length later than when it actually occurred. This is easily remedied in a spreadsheet by shifting the ordinate data by one-half of the window length,[6] and in the case of fast data acquisition such a time shift is mostly likely to be negligible.

The implicit precautionary principle here is illustrated by the work of Felinger et. al,[7] where a slow DAQ (down to 1.25 Hz) is used, and noise is smoothed by a process involving trigonometric interpolation. This process carries its own shortcomings, but from equation (3.1.4.1) above, a slower DAQ does lend itself to significant phase (and so also signal) shift when using window-based smoothing.

A second precautionary principle is that the processing algorithm **Ξ,** has underlying assumptions such as those listed in § 3.1 (i) – (iii) above, and this may have the consequence of making absorbance peaks appear more or less efficient than they really are.[8] This may happen because some functions underlying an algorithm have either natural minima such as a cubic or quartic functions, asymptotic tendency such as exponential or logarithmic functions, or even by other methods such as discrete wavelet transform (DWT).[9] Some of the algorithmic possibilities and their outcomes are discussed further in Chapter 4.

Smoothing algorithms in signal analysis in CE or LC make use of windows of point clusters of different sizes in order to achieve smoothing of differing degrees. The growth of interest in Miniaturised Total Analysis Systems (μTAS) which run on small commercially available and cheap microprocessors[10] means that optimised algorithms need to be found to enable smoothing of incoming signals in real time in order to simultaneously achieve economy of processing time and smoothing.

For example, in an instrument such as the Agilent 35900E which uses a RS-232 microchip with maximum data transfer rate of 38400 baud with a 24-bit analog/digital converter as described in Chapter 2, the parameters of Savitsky-Golay smoothing can be controlled by using the menu to adjust the number of terms or the order of the approximating polynomial.

Also, for commonly used optical absorbance detectors, choices embedded in detector electronics may allow a user to choose a particular data sampling frequency in Hz together with a response time or a time constant, but variance contributions from detector electronics may then have an impact on the peak height, peak width and symmetry.[11]

3.2 Short General Discussion on applicable Noise Theory

An analog-output CE or LC device which needs to detect and measure very low concentrations at high sampling rate will have limited interaction with the molecules it is intended to sense, (with some exceptions such as cell counters which operate in the positive integer domain) and when signal magnitude is of the same order of magnitude as noise, precision is likely to be poor. Some limiting factors on precision and accuracy in using absorbance measurements to determine low analyte signals against the constraint of a high noise background are:

(i) Thermal (kinetic) fluctuations and charge;
(ii) Quantisation of light and energy;
(iii) Inconsistency in ADC due to rounding errors;
(iv) Flicker, or $\frac{1}{f}$ noise.[12]

This latter category of *flicker noise* is a generic category because the source of what appears to be a random effect is mostly unknown. What seems to be the case is that it

does contain some underlying patterns, because it can be simulated by using fractals and nonlinear dynamics.[13]

Variations in light output from a LED (particularly operating near the lower or upper limit of its operating voltage) or thermo-luminescent lamp is an example of $\frac{1}{f}$ noise, which becomes more pronounced when DAQ < 300 Hz. This appears to be true even when light in the transducer is highly coherent.[14] Such noise may be reduced by modulating the signal through an amplifier.

Such factors cause random signal fluctuations and the cumulative effect of factors such as those outlined in (i)-(iv) above is collectively known as *noise* in analytical vernacular, on the understanding that it has its origin in different parts of an instrument. An illustration of such distinctions appears in Figure 3.2.1 below.

Noise is an inverse measure of precision, and so effort is needed to reduce its influence on the signal. Measurement devices may be miniaturised, but transducer size is often limited by the physical parameter which it measures; in electrophoresis, the transducer is mostly a sensor which converts light energy or conductivity into electrical potential difference.

There are many ways of achieving this; for example, a simple average of measurements over time may reduce noise. For static discrete measurements or measurements which are very slow to change with respect to the sampling rate, a good measure of the Y-direction (ordinate) noise is the standard deviation:

$$\sigma_A = \sqrt{\frac{\sum_{j=1}^{2i+1}(A_j - \mu_A)^2}{(2i+1)}} \qquad (3.2.1)$$

Where j is a counter variable;

A_j are the ordinate data points;

$2i+1$ is the (odd) number of points in the smoothing window; and

μ_A is the simple average of the ordinate data over window size $2i+1$.

Since we are dealing with electropherograms where noise fluctuations are overwhelmingly within the ordinate (Y-axis), data point readings are attributed the symbol A_i which also assumes that sampling occurs at consistent uniform time intervals, where

$$\forall i, n \in \mathbb{Z}_n^+, i < n, t_i - t_{i-1} = t_{i+1} - t_i \qquad (3.2.2)$$

(For all ($\forall$) counter values i which are positive integers below the cardinal number n of the entire dataset, the time differences between any three successive readings t_{i-1}, t_i and t_{i+1} is constant.)

In the case of fast data acquisition rate, it turns out that such an assumption is not reliable, and it turns out that $t_i - t_{i-1} \neq t_{i+1} - t_i$ for an increasingly significant proportion of the dataset as DAQ increases above 600 Hz. In this discussion, time variability (X-direction or *abcissa* noise) will be referred to as *jitter*.[15]

This immediately means that the simple ordinate standard deviation σ shown above must be modified to include jitter in a 2-dimensional (root mean square RMS) analysis of noise σ_{yx}. In a spreadsheet, this is easily done by invoking the STEYX function, which takes account of both Y and X direction (ordinate and abscissa) noise.

$$\sigma_{yx} = \sqrt{\frac{1}{(2i+1)} \sum_{j=1}^{2i+1} \left[(y_j - \mu_y)^2 + (x_j - \mu_x)^2 \right]} \qquad (3.2.3)$$

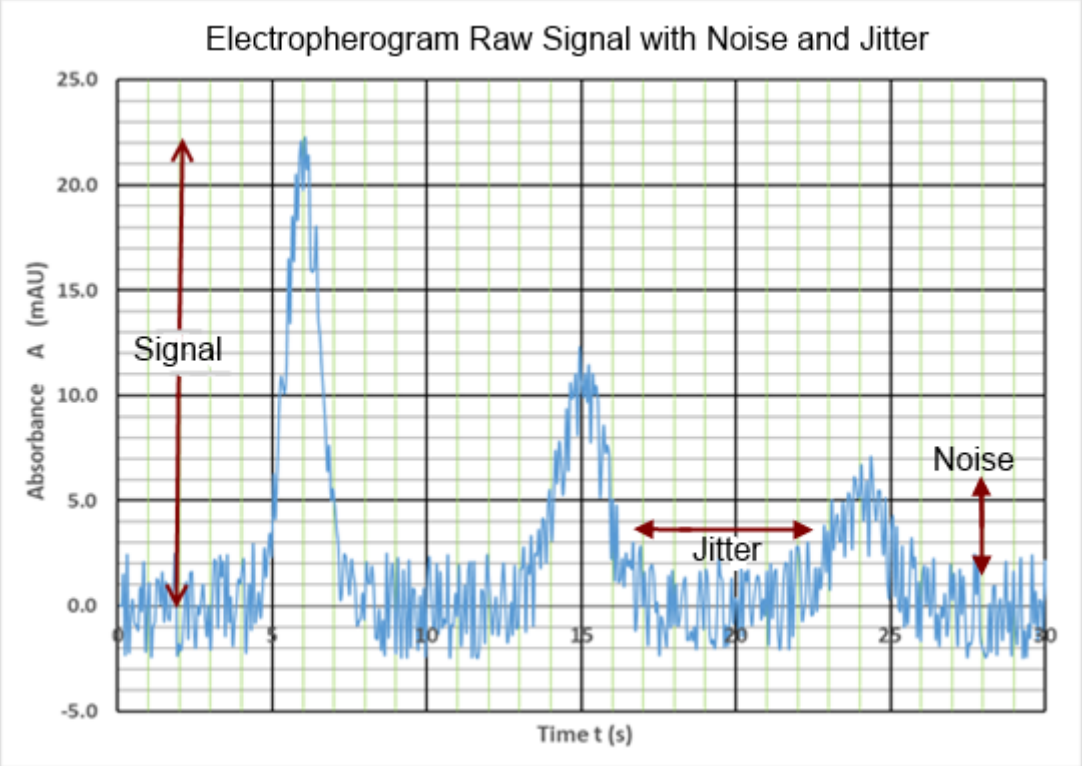


Figure 3.2.1: Constructed electrophoretic compound dataset to illustrate the definitions of signal, jitter and noise as they appear in fast data acquisition discussed above. The Absorbance A is measured in appropriate Absorbance Units (AU)
DAQ= 315Hz. For Noise alone: $\sigma_A = 1.46$; For Noise *and* Jitter: $\sigma_{At} = 1.57$

The notation now changes from σ_{yx} to σ_{At} to denote absorbance A as the ordinate (Y) variable and time t as the abcissa (X) variable.

When this is applied to the dataset (t_i, A_i), it leads to the use of the STEYX function in Microsoft Excel™ being applied in the smoothing algorithms, by anchoring the calculation of σ_{At} at the central $(i+1)^{th}$ point, and averaging the 2-dimensional standard deviation of the two half-windows of i data points on either side:

$$\sigma_{At} = \frac{1}{2i}\left\{\sqrt{\sum_{j=1}^{i}\left[\left(A_j-\mu_A\right)^2+\left(t_j-\mu_t\right)^2\right]}+\sqrt{\sum_{j=i+2}^{2i+1}\left[\left(A_j-\mu_A\right)^2+\left(t_j-\mu_t\right)^2\right]}\right\} \quad (3.2.4)$$

(Where μ_A and μ_t are the respective means of A and t over each half of the smoothing window.) This difference between one and two-dimensional has a subsequent effect on SNR.[16]

This has a direct effect on the first derivative method for initial identification of signals,[17] which is a technique which is often used. It affects this technique, which will be used in Chapter 5 because the immediate consequence of a high frequency and high noise (lowered SNR) sampling rate is that the first derivative of the initial electropherogram will reflect and exaggerate both noise and jitter, as was rigorously predicted by Levant.[18] This is illustrated in Figure 3.2.2; and so becomes of little use for a visual guide to either migration time or peak width.

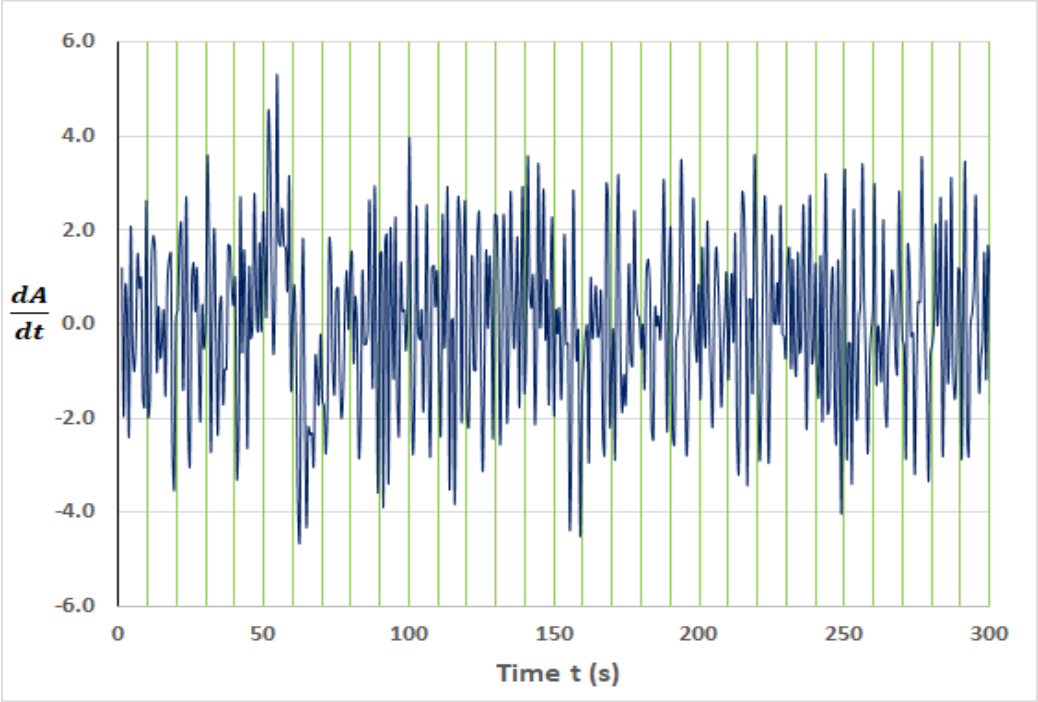


Figure 3.2.2: First derivative of the dataset illustrated in Figure 3.2.1 above.

Determination of peak width, migration times and other parameters through common numerical methods applied to the first derivative of a noisy signal is also unreliable, yielding high levels of uncertainty.

3.2.1 Baseline Effects of two-dimensional noise.

Signal-to-noise ratio SNR has no meaning without first establishing a baseline, and the determination of baseline in a high-noise/fast DAQ means that a first analysis of noise must be comprehensive if a low SNR signal is to be of analytical usefulness. The work done in §3.2 above on two-dimensional noise can now be applied to baseline determination. The consequence of this shift from single dimension σ_A to the two-dimensional version σ_{At}, is that any intuitive assumption that the chromatograph baseline lies around the mid-point of ordinate (Y-direction) noise, is no longer reliable. For fast DAQ, low resolution ADC, high noise and low SNR especially this need not be the case.

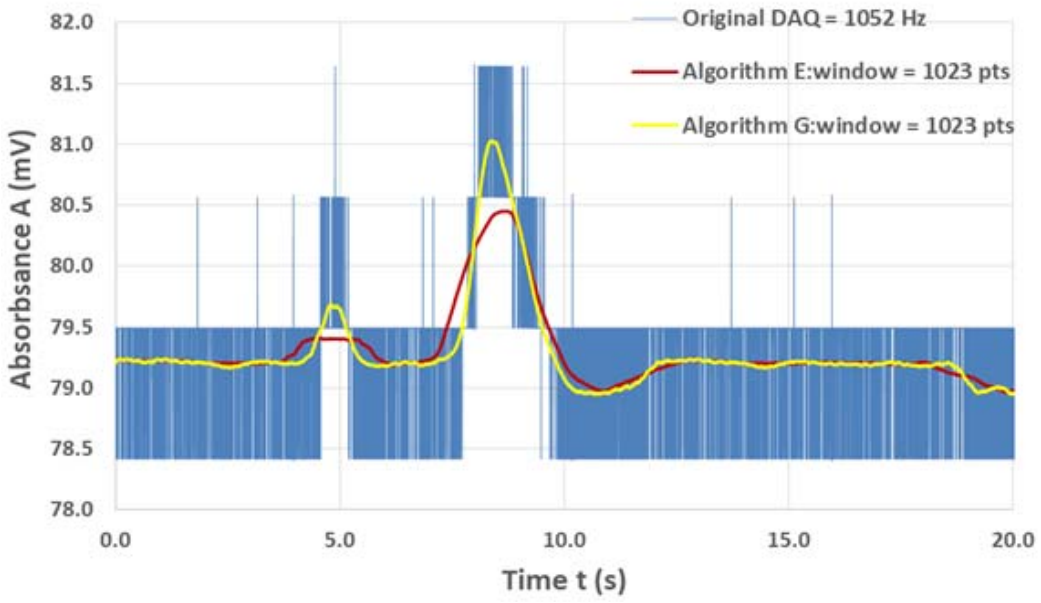


Figure 3.2.1.1: Dataset to illustrate the position of baseline after application of two different smoothing algorithms to fast data acquisition at low resolution ADC with significant jitter. DAQ = 1052 Hz. For ordinate noise only: $\sigma_A = 2.01$; For Noise and Jitter: $\sigma_{At} = 3.46$

At this juncture it is necessary to refer to the measurement of time intervals during DAQ shown in Figure 3.2.1.1. In the case of the Arduino UNO – based instrument used in this current discussion and described in Chapter 2, the time intervals used were accessed by the *mills()* function, shown in the full coding of the operating system in APPENDIX 2.

A smoothed baseline signal may be weighted towards upper or lower regions of the digitised values of raw data, depending on the localised density of the high or low values assigned by the ADC chip, as raw data determines the low or high value of the subsequent smoothed signal. If, for example, we extract regions around 5.0 s and 15.0 s from Figure 3.2.1.1, these trends are further illustrated in Figure 3.2.1.2.

There is a lag of approximately 0.5 s in the effect because a window of 1023 points means that $i = 511$ and the lag (or phase shift) is then

$$\varphi = \frac{i}{DAQ} = \frac{511}{1023} \cong 0.5 \text{ s}$$

from equation 3.1.4.1 above.

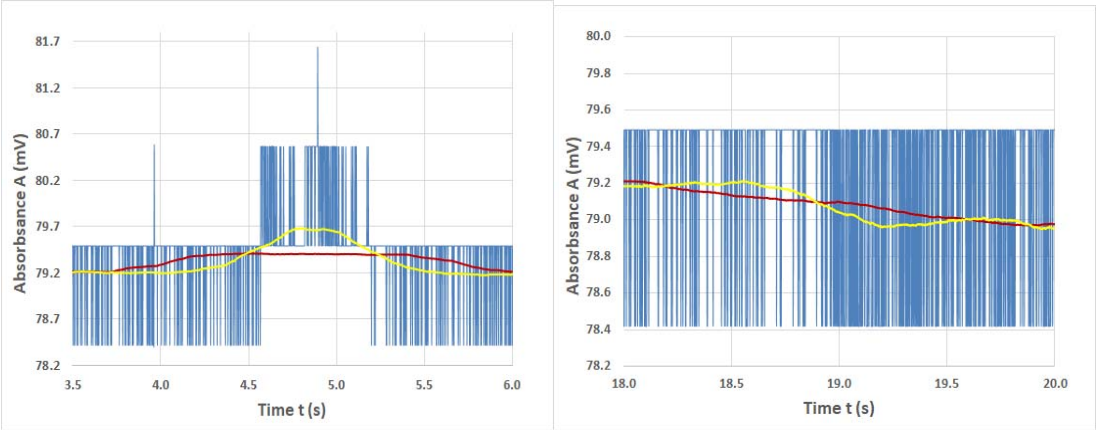


Figure 3.2.1.2: Extracts from Figure 3.2.1.1 to illustrate delayed baseline fluctuation with data density after application of smoothing algorithms to fast data acquisition with significant jitter.

What also becomes apparent in §§3.3.1 – 3.3.6 below is that not all algorithms are equally responsive to rapid changes in digitised high/low data distribution.

When data acquisition rate is in excess of several hundred Hz, the data reading and processing by the chip operating system (however efficient and elegant the algorithmic program may be) is a non-zero component of the maximum acquisition rate, and so uncertainties in time intervals become significant; this accounts for an increase in standard deviation from σ_A to σ_{At}. If the time intervals were to be averaged over larger intervals (such as 1.0 s or even 2.0 s) then two consequences follow:

- the direct connection between individual absorbance values A_j and their corresponding time values t_j are lost;

- the averaging of the absorbance values A_j must necessarily follow, which means that a segment of the data has already been artificially smoothed before any subsequent algorithm is applied.

Table 3.2.1.1: Comparisons of standard deviations from σ_A to σ_{At} for the original datasets of Figures 2.3.1.1, 3.2.1, 3.3.1 and 3.2.21.1

	DAQ (Hz)	σ_A	σ_{At}	% Increase
Fig 2.3.1.1	10	0.77	0.77	0
Fig 3.2.1	315	1.46	1.57	7.53
Fig 3.3.1	575	1.87	2.48	49.1
Fig 3.2.1.1	1052	2.01	3.46	72.1

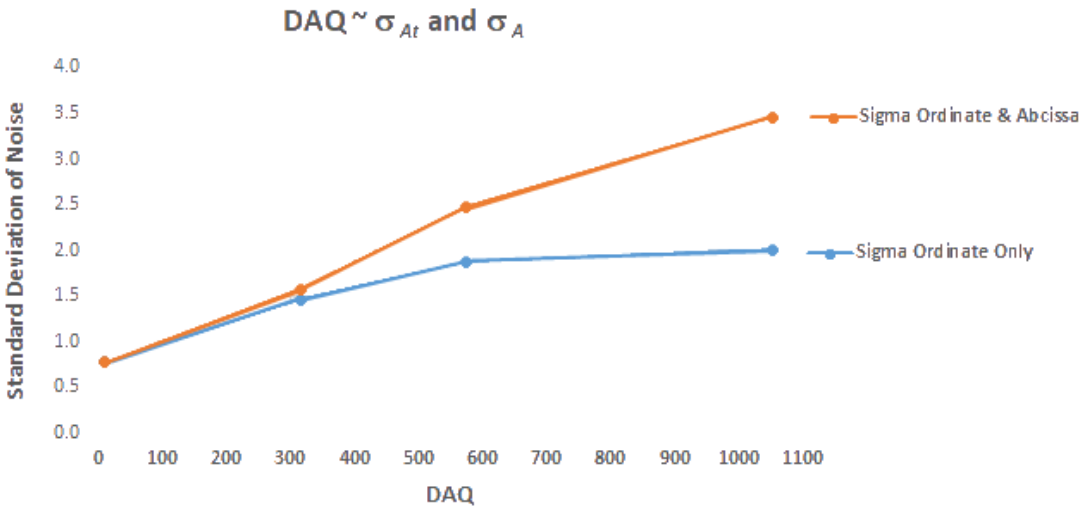


Figure 3.2.1.3: Difference in standard deviations for one dimension vs two dimensions with DAQ. Increase of σ_{At} over σ_A only is due to the added influence of jitter at high DAQ.

Two lemmata (See **List of Abbreviations and Definitions** for definition) are now stated here without proof:

Lemma 1: When jitter as measured by $\Delta t \rightarrow 0$ at low DAQ, then $\sigma_{At} \rightarrow \sigma_A$.

Lemma 2: When $\Delta t = 0$, then $\sigma_{At} = \sigma_A$.

(These lemmata can be derived from the discussion around equation 3.2.2 above, and the application of limits to equation 3.2.4.)

3.2.2 SNR Effects of two-dimensional noise

Since SNR is a function of both signal magnitude above an average baseline, and neighbouring noise as measured by either σ_A or σ_{At}, we can write

$$SNR = \frac{A_{max} - (\frac{A_{left} + A_{right}}{2})}{\sigma_{At}} \tag{3.2.2.1}$$

Where:

(t_{left}, A_{left}) and (t_{right}, A_{right}) are the left and right turning points of a signal;

(t_{max}, A_{max}) is the maximum point of a signal. This is easily determined in a list of data by the use of a line of software.

These features are illustrated in Figure 3.2.3.1 below.

As a consequence of the lemmata 1 and 2, it always follows that:

$$\sigma_{At} \geq \sigma_A \tag{3.2.2.2}$$

This increase necessarily decreases SNR, allowing smoothing algorithms to be studied in a way which takes account of both ordinate and abcissa noise without a possible *a priori* distortion such as that discussed just prior to Figure 3.2.1.2 above.

3.2.3 Some Notes on Symmetry/Asymmetry: A Novel Determination of Peak Width, and hence baseline and Peak Height

The obvious trouble with smoothing algorithms is that by the imposition of mathematical order on the chaos of noise, they impose a pre-determined shape on the data. For example, typical electropherograms in capillary electrophoresis are not Gaussian-type symmetrical, but triangular; mostly having sharp initial gradient and a much more gradual back tail, or gradual initial gradient and sharp back due to electromigration dispersion. The choice of smoothing algorithm is mostly out of the hands of the experimenter either because it is hard-wired into the instrument, or the choices are limited by the manufacturer.

Signal asymmetry raises questions about the reliability of finding the turning points at the base of the signal of interest, and so introduces uncertainty into peak width. One technique which sidesteps this uncertainty is to find peak width at half-height, where the asymmetry of peak is less pronounced. This technique has the advantage of halving the uncertainty; however the question of peak height from the baseline remains, because determination of turning points – both from baseline to peak and at the peak itself are further possible sources of uncertainty.

A way of addressing this issue is to adopt the following technique using the length of a vector.

- The peak of the signal of interest defines a unique migration time, and this point can be designated as (t_{max} , A_{max}) as illustrated in Figure 3.2.3.1 below.
- A key point to anchor the vector is selected at around the value of t_{max} (certainly between t_{left} and t_{right}) and about half the peak height below the perceived baseline. Call this anchor point (t^*, A^*).
- A vector $\vec{l}$ from (t^*, A^*) to any j^{th} point (t_j , A_j) on the data has length $|\vec{l}|$ given by

$$|\vec{l}| = \sqrt{(t_j - t^*)^2 + (A_j - A^*)^2} \qquad (3.2.3.1)$$

The key is that there is only one shortest length for such a vector in any interval of data points; in particular, between (t_j, A_j) and (t_{max}, A_{max}) [$j < max$] and then again between (t_{max}, A_{max}) and (t_j, A_j) [$j > max$].

Experimentally, it is found that the best results are obtained when

$$5\times|t_{max}| \leq |A_{max}| \leq 10\times|t_{max}|$$

(The absolute value of A_{max} should be more than $5\times|t_{max}|$ but less than $10\times|t_{max}|$)
Outside these limits the representation of $|\vec{l}|$~t becomes increasingly difficult for visual inspection because the curve has very low amplitude or very high amplitude to be of much visual use. A scaling factor has no effect on the maxima, minima or zeros of the function.

A plot of $|\vec{l}|$~t is of no use in a noisy signal, because noise in the regions around (t_{left} , A_{left}) and (t_{right} , A_{right}) remain noisy and so the point values themselves are difficult to determine. (This is illustrated in Chapter 5, Figure 5.2.1.8)

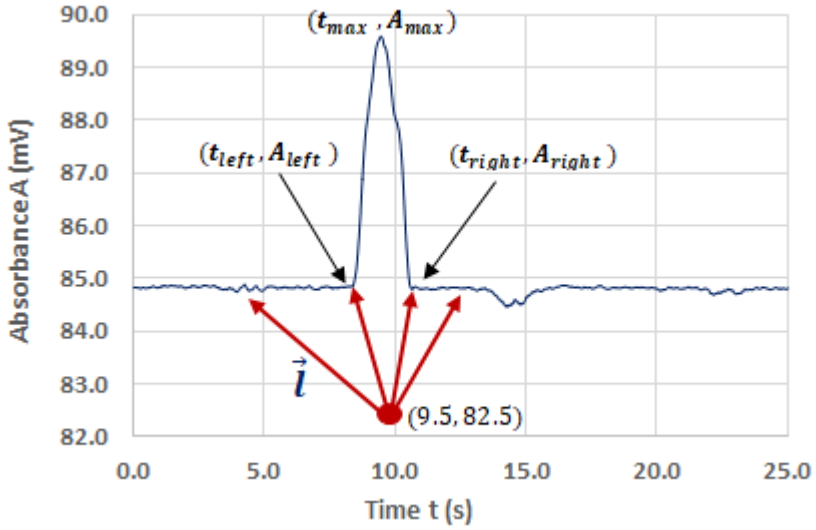


Figure 3.2.3.1: Illustration of the vector method for determination of a peak width at baseline. The anchor point (t^*, A^*) is the point (9.5 , 82.5)

The technique discussed here however focuses on the fact that there are *two* determinants of these points, viz:

- A minimum length of the vector within each interval (t_{left} , t_{max}) and (t_{max} , t_{right})
- The instantaneous change in the trend of the length $|\vec{l}|$ as it decreases from (t_j, A_j) to (t_{left} , A_{left}) and then instantaneously increases between (t_{left} , A_{left}) and (t_{max} , A_{max}); the same happens between (t_{max} , A_{max}) and (t_{right} , A_{right}) and further to (t_j, A_j)

In particular, if one begins with

$$|\vec{l}| = \sqrt{(t_j - t^*)^2 + (A_j - A^*)^2}$$

then equation 3.2.3.1 above can be re-written as:

$$|\vec{l}| = \left[(t_j - t^*)^2 + (A_j - A^*)^2\right]^{\frac{1}{2}}$$

Whence

$$\frac{d|\vec{l}|}{dt} = \frac{(t_j - t^*) + \frac{dA}{dt}(A_j - A^*)}{|\vec{l}|} \qquad (3.2.3.2)$$

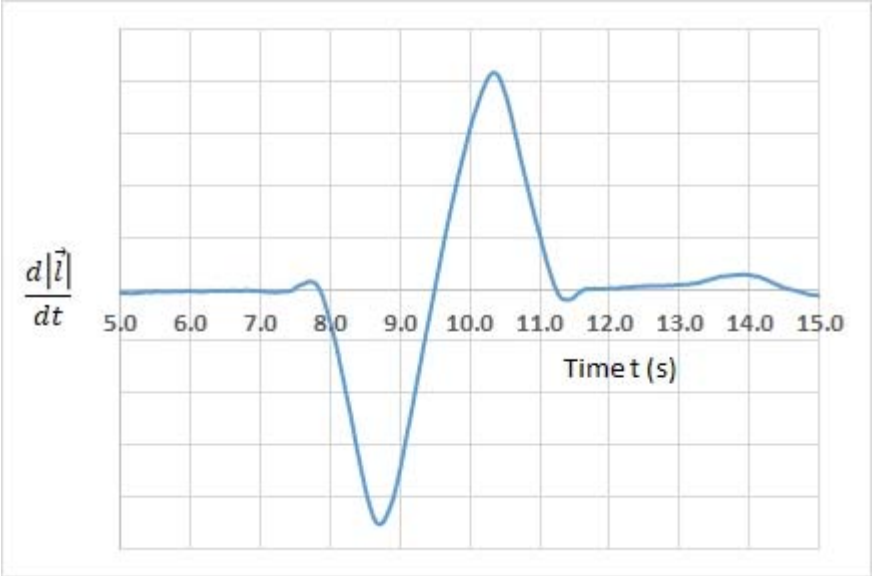


Figure 3.2.3.2: Illustration of the differential plot of equation 3.2.3.2 from the vector method for determination of a peak width at baseline.

An application of data functions such as VLOOKUP, INDEX, MAX and MIN in Microsoft Excel™ allow for unambiguous determination of (t_{left} , A_{left}) , (t_{max} , A_{max}) and (t_{right} , A_{right}) from the function in Figure 3.2.3.2 above. Such a spreadsheet can be viewed in APPENDIX 3.

In all electropherograms hereafter, including Chapters 4 and 5, this technique is used to determine peak width at the baseline. Migration times are corrected for smoothing window size by time-shifting the data after the method of Dasgupta.[4]

3.3 Experimental Outline

To study the effects of smoothing algorithms on real data which is both noisy and acquired at fast DAQ, an inexpensive 10-bit ADC running at variable data acquisition rates of up to about 2.0 kHz was used to acquire electrophoretic data in two cases.

(i) The first case used is the detection of fluorescein after hydrodynamic injection with fluorescein sample driven between buffer plugs, using the apparatus described in Chapter 2;

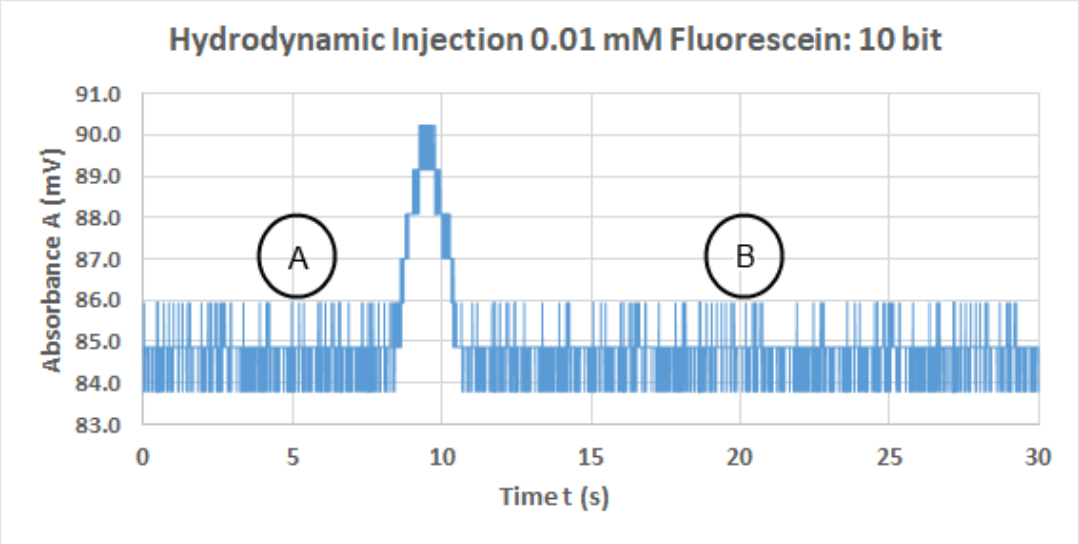


Figure 3.3.1: Original Dataset for electrophoretic detection of fluorescein. For this Arduino-controlled system, a fused silica capillary is used with L_D = 12.5 cm; L_T = 15.0 cm; V= 6.0 kV; E=400.0 V.cm^{-1}; 25 µm i.d. and 363 µm o.d. with standard polyimide coating. DAQ = 575 Hz.

Region A: $\sigma_A =$ 1.89 Region B: $\sigma_A =$ 1.86; Weighted average $\sigma_A =$ 1.87

Region A: σ_{At}= 2.50 Region B: σ_{At}= 2.47; Weighted average σ_{At} =2.48

Initial comparisons were made between five common mathematical smoothing algorithms used in chromatographic analysis against an application of the Nyquist Theorem, beginning with the single-peak dataset shown in Figure 3.3.1 above.

There is no pre-judged order of importance for these algorithms, and so the strategy employed was simply to take them in an arbitrary order of perceived increasing complexity as follows:

- Boxcar (§ 3.3.1)
- Geometric Mean Smoothing (GMS) (§ 3.3.2)
- Exponential Mean Smoothing (EMS) (§ 3.3.3)
- Savitsky-Golay (S-G) (§ 3.3.4)
- Gaussian Smoothing (§ 3.3.5)
- Nyquist (§ 3.3.6)

Some principles and trends from these comparisons began to emerge, and were subsequently tested by using a second electropherogram of separated chromophores illustrated in Figure 3.3.2 below.

(ii) This second case shows the electrophoretic separation of fluorescein and fluorescein isothiocyanate (with Coumerin-102 to show the position of the EOF) again with fast data acquisition, as discussed in Chapter 2. This second case also allowed for the small EOF to be discerned from the noise, and also enabled some analysis of resolution and efficiency.

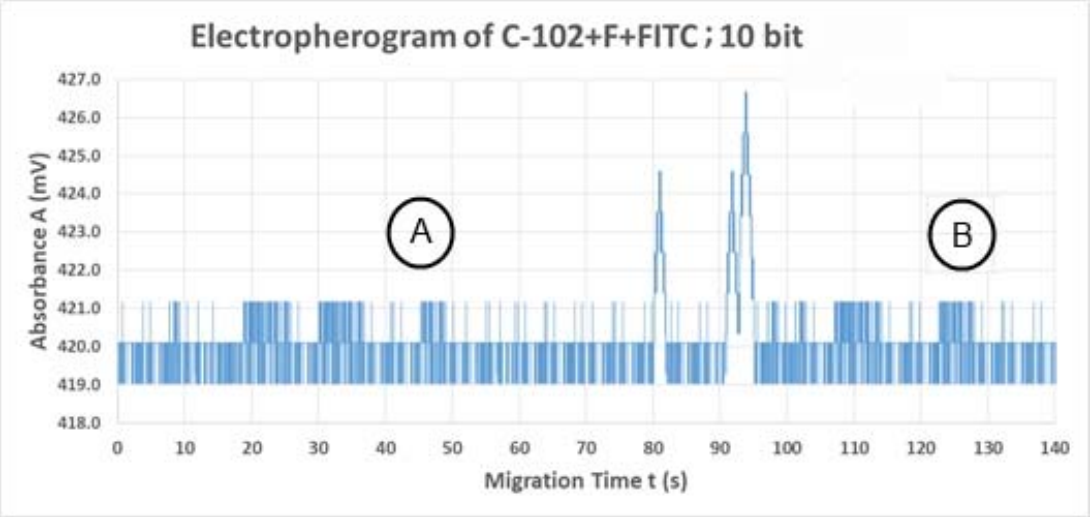


Figure 3.3.2: Original Dataset for electrophoretic separation of fluorescein, fluorescein isothiocyanate and coumerin-102. For this Arduino-controlled system, a fused silica capillary is used with L_D = 12.5 cm; L_T = 15.0 cm; V= 6.0 kV; E=400.0 V.cm^{-1}; 25µm i.d. and 363µm o.d. with standard polyimide coating. DAQ = 875Hz.

The original datasets in Figures 3.3.1 and 3.3.2 were acquired at DAQ rates of 575Hz, and 875Hz respectively - much less than the maximum DAQ of the Arduino UNO, but which can still be classified as “fast”. The discrepancy seen as time lag is due to two factors:

- Firstly, the analog data is fed into the Arduino UNO where it is converted via the ADC to a mV value through the C++ operating system as shown in APPENDIX 2;
- Secondly, this output is fed into a second piece of software entitled "Processing", which takes this digitised data from the Arduino UNO, formats it as a .csv file which is then written continuously into a spreadsheet in real time. At these rates, both signal and noise are sampled without discrimination.

In subsequent sections of this chapter, the algorithms are applied to the data *post-facto*, and the effect of each algorithm at different window sizes is tested for SNR, peak width and resolution to determine optimum window sizes. Some conclusions are then drawn with respect to the applicability of various algorithms at different window sizes.

3.3.1 The Boxcar: Moving Average Algorithm or Rectangular Algorithm

The simplest software method for smoothing an array of raw and noisy data into a new array of smoothed data is where each smoothed point (t_{ks}, A_{ks}) is the unweighted average of an odd number $2i+1$ ($i = 1, 2, 3, ..$) of consecutive points of the raw data centred at the K^{th} term:

$A_1, A_2, A_3, \ldots\ldots\ldots\ldots A_{2k+1}$ such that

$$(t_{ks}, A_{ks}) = \left(\frac{1}{2i+1} \sum_{j=1}^{2i+1} t_j \ , \frac{1}{2i+1} \sum_{j=1}^{2i+1} A_j \right) \tag{3.3.1.1}$$

Where j is the counter.

The larger this smoothing window of $2i+1$ points becomes, the more pronounced the smoothing effect. As will be seen in Figure 3.4.1.1 and Figure 3.4.1.2 below, the smoothing is initially very rapid, but as filter width increases beyond 300 points, the SNR decreases with peak widening and produces a more pronounced lag in the smoothed sequence.

There are three disadvantages to this technique:

- it cannot be used on the first i terms of the series without the addition of values as described in §3.1.4;
- it averages the times, and so each completed window of smoothing consisting of $2i+1$ points becomes less weighted by data density through loss of resolution;

- the rectangular (ordinate and abcissa) effect of the boxcar is to reduce time differences between the centre k of each successive smoothed cluster, and so when $\Delta t \rightarrow 0$, then $\sigma_{At} \rightarrow \sigma_A$ by Lemma 1, and any correction of SNR due to jitter is reduced.

3.3.2 Geometric Mean Smoothing (GMS)

This method for weighted mean smoothing for a set of raw data (t_k ,A_k) is to calculate a weighted moving average by the two following simultaneous conditions:

(i) Choosing an odd number $2i+1$ of terms, based on i weighting factors $w_1, w_2, w_3, \ldots\ldots\ldots\ldots w_{2i+1}$, such that

$$\sum_{j=1}^{2i+1} w_j = 1$$
$$\text{and} \quad w_j \in \{(0,1) \subset \Re^+\} \qquad (3.3.2.1)$$

AND

(ii) Allocation of weightings is such that the highest weighting is assigned to the central anchor k^{th} term (w_k)and weightings decrease symmetrically to positions 1 and $2i+1$ respectively from w_k

$$(t_{ks}\,, A_{ks}) = \sum_{j=1}^{2i+1} (w_j t_j\,, w_j A_j) \qquad (3.3.2.2)$$

Two advantages of this technique are that weighting factors give more weight to the most recent terms in the time series and less weight to older data, and as a consequence, there is less flattening of signals as will be shown below. Secondly, equation 3.3.2.2 above shows that the technique takes jitter into account, which rectangular smoothing does not.

The disadvantages of this technique are:

- Again that it cannot be used until at least $k = 2i+1$observations have been made;
- If the GMS smoothing algorithm is embedded in the software, it entails a more complicated calculation at each step of the smoothing procedure than boxcar. This is due to the necessary insertion of a call procedure in the software, where the weighting constants ${}^{2i+1}_{j=1}w_j$ will need to be called from a separate matrix and assigned to each point successively in the window. The result of these steps is that DAQ is decreased.

3.3.3 Exponential Mean Smoothing (EMS); Exponentially Weighted Moving Average (EWMA); Autoregressive Integrated Moving Average (ARIMA)

Exponential Mean Smoothing (EMS) applies the Poisson window function, the use of which goes back to the 1940s. In this study, the formulation of equation 3.3.3.1 below is the basis of the method used, which has been modified in this study from the methods of Holt and Winters[19] to facilitate recursive filtering and allow for less cumbersome adaptation to programming in C++ and subsequent data analysis from a spreadsheet.

The subscript "s" refers to the smoothed value, and the simplest form of exponential smoothing is given by the following formula:

When $k = 1$, then the first point cannot be smoothed, and so $A_{ks} = A_{1s}$ and $t_{ks} = t_{1s}$

$(t_{1s}, A_{1s}) = (t_1, A_1)$ for $k=1$, and the first point is unchanged; subsequently
$A_{ks} = \alpha A_k + (1 - \alpha) A_{k-1}$ and
$t_{ks} = \alpha t_k + (1 - \alpha) t_{k-1}$ where $k \geq 2$, $k \in \mathbb{Z}^+$
and α is the smoothing factor where $\alpha \in \{(0,1) \subset \Re\}$

(3.3.3.1)

Again, such a process deals with jitter and the time-associated signal noise in the same way. The smoothing factor α is obtained by choosing it through the Levenberg–Marquardt algorithm coded into Microsoft Excel™ Solver[20]. This chooses a value of α which minimises the mean square error (MSE).

For the single-peak electropherogram of Figure 3.3.1, $\alpha = 0.591$

For the multiple-peak electropherogram of Figure 3.3.2, $\alpha = 0.537$

The smoothed value $(t_{ks}, A_{ks})$is a processed and weighted average of (t_k, A_k) and the previous smoothed statistic (t_{k-1}, A_{k-1}). The interval (0,1) is open, since

(i) if $\alpha = 1$ the output series is just the same as the original series, because as $\alpha \to 1$, there is less of a smoothing effect, with greater weight to recent changes in the data; and

(ii) as $\alpha \to 0$, smoothing effect is greater and data points are less responsive to recent changes - eventually if $\alpha = 0$, the smoothed values are all identical, and the outcome is trivial.

The programming difficulty is that each successive iteration requires a recalculation of the previous one by equation 3.3.3.1, which means that unless software shortcuts are introduced, data smoothing becomes progressively slower as it progresses through the dataset. The modified exponential smoothing[21] used in this study[22] can be easily applied in a spreadsheet, and as usual can produce a smoothed statistic as soon as at least three observations are available.

If there are some missing data values, the original signal may be approximately reconstructed. For best reconstruction by smoothing of an original signal with minimal information loss, all time values of the exponential moving average must be available[23], which will show the number and positions of missing data points. Signal reconstruction depends on reverse application of equation 3.3.3.1, as previous samples decay exponentially in weighting. Similar conditions apply to some of the other algorithms.

3.3.4 The Savitsky-Golay (S-G) Filter

The Savitzky–Golay method is applied to digital data points in sets of increasing size to effect smoothing and to increase SNR with minimal distortion of either peak width or resolution. This is achieved by best-fitting successive odd-numbered subsets of data points with a low-degree polynomial using linear least squares. Data consists of a set of n points (t_j, A_j) where $j = 1, 2, \ldots\ldots\ldots n$ and where the t_j is the independent (time) variable and A_j is the absorbance value at time t_j. Conventionally, when the t_j are equally spaced, an analytical solution to the least-squares equations can be found in the form of a single set of convolution coefficients[24] applicable to any odd-numbered data subset to give a best estimate of a smoothed signal (or subsequent derivatives of the smoothed signal) around the central point of each smoothing window. (This central point requirement is the reason for the restriction to odd-numbered subsets.) In the modern context of electronic data acquisition, this is true (at least with a very small margin of error) where data acquisition rates are slow (up to about 50Hz) as they were when Savitsky and Golay wrote the seminal paper[25] in 1964, which was later corrected in the famous letter by Madden.[26]

The data set of odd-numbered cluster of p points centred at position m, so that

$j = 1, 2, \ldots, (m-1), \mathbf{m}, (m+1), (m+2), \ldots\ldots p.$

Note: A cluster is not the same as a smoothing window; the cluster refers to the number of points which are fitted by the best-fit polynomial to smooth the signals within the cluster. For example, a 7-point cluster will take 7 consecutive points and replace them all with a new set of points fitting as closely as possible to a 7-point polynomial.

The centre point of the cluster moves from one end of the window to the other, centred at the anchor point of the window. Smoothing within the window then occurs in a successive series of clusters. It is a necessary condition that $p < 2i + 1$. The S-G polynomial cluster may not be greater than the smoothing window; for example we cannot have a window of 3 points smoothed by a 7-point polynomial.

This is illustrated with a concrete example in Figure 3.3.4.1 below.

Smoothing window of k=37 points centred at point 19. So $i = 18$ [1 ,2, 3, 4, 5, ……16, 17, 18, **19**, 20, 21, 22, ……33, 34, 35, 36, 37]
(1, 2, **3**, 4, 5) → (9, 10, **11**, 12, 13) → (17, 18, **19**, 20, 21) → (25, 26, **27**, 28, 29) etc, then (2, 3, **4**, 5, 6) → (10, 11, **12**, 13, 14) → (18, 19, **20**, 21, 22) → (26, 27, **28**, 29, 30) etc Polynomial cluster of $p = 5$ points moves successively from one end of the smoothing window to the other.
The smoothing window of 37 points now shifts one point to be centred at point 20. [2, 3, 4, 5, 6, ……., 17, 18, 19, **20**, 21, 22, 23,…… 34, 35, 36, 37, 38] And the process continues in the same two steps through the entire dataset.

Figure 3.3.4.1: Schematic presentation of Savitsky Golay cluster progressing through a smoothing window of data points.

Each of the p points in the cluster are treated with a set of integer convolution coefficients, C_j, where A_m is the centre of the cluster, and $p = 2m - 1$

$$A_m = \left(\frac{1}{\sum_{j=1}^{p} C_j}\right)\left[C_m A_m + \sum_{j=1}^{m-1} C_j\left(A_{m-j} + A_{m+j}\right)\right]$$

$$A_m = \frac{1}{h}\left[C_m A_m + \sum_{j=1}^{m-1} C_j\left(A_{m-j} + A_{m+j}\right)\right] \qquad (3.3.4.1)$$

$h = \sum_{j=1}^{p} C_j$ is the *integer normalisation constant*. For convenience, this normalisation constant is calculated for the chosen set of convolution coefficients prior to the application of the data smoothing process, to speed up the software program.

The SG procedure performs a least squares fit of a *small* subset where ($p << n$) of consecutive data points to a polynomial and the calculated central point A_m of the fitted polynomial curve becomes the new smoothed data point. Convolution integers are weighting coefficients to carry out the smoothing operation, exactly equivalent to fitting the data to a polynomial, and it is computationally more effective and faster.

(The convolution coefficients are shown in APPENDIX 3, Table Apx 3.2.)

To summarise: The steps can be illustrated by thinking of a sequence of logical steps in a computer programme using pseudo-code:

1. Pick a cluster size e.g. $p = 7$;
2. Pick a window size e.g. $n = 59$;
3. Centre the cluster at window point 4 and apply polynomial smoothing to first 7 points;
4. Move the centre of the cluster to point 11 and repeat until centre reaches point 56;
5. Move window centre from point 30 to point 31;
6. Repeat stepwise from step 4 until you run out of data.

In this study, it was decided to use a 5-point polynomial (polynomial degree = 4) because in real time, a program which additionally uses a compound of two algorithms and processes data *in situ* (as discussed subsequently in Chapter 4) is significantly slowed by polynomials of degree greater than 4. Such a formula is also amenable to spreadsheet application.

In this concrete example: For smoothing by a 5-point polynomial;

$j = -2, -1, 0, 1, 2$; and the m^{th} smoothed data point, A_{ms}, within the cluster is given by:

$$A_{ms} = \frac{1}{35}[-3 \times A_{m-2} + 12 \times A_{m-1} + 17 \times A_m + 12 \times A_{m+1} - 3 \times A_{m+2}]$$

And $h = (-3)+(12)+(17)+(12)+(-3) = 35$

3.3.5 Gaussian Smoothing

The Gaussian kernel is one of the most widely used smoothing kernels, and is expressed in adapted form by

$$G_j(A_k, A_j) = exp\left[-\frac{(A_k - j)^2}{2(t_k - t_j)^2}\right] \quad (3.3.5.1)$$

Where as usual, $(A_k\,,\, t_k)$ is the central point of the Gaussian window, and $(t_k - t_j)^2$ is the length scale for the input space.

What immediately becomes clear is that this is a two-dimensional statistical representation which takes account of jitter. From Figure 3.4.1 below, the inherent assumption of the Gaussian fit can be seen intuitively, and it turns out that the greater the value of k (window size) the more closely the peak shape approaches an ideal Gaussian shape.

The basic process of Gaussian smoothing with the kernel, is again to proceed pointwise through the data. For each data point the algorithm generates a new value that is a Gaussian function of the original value at that point and the k data points on either side. Each successive point is the centre of the Gaussian curve. The integral of this Gaussian curve is the area under the curve.

To ensure that smoothed data does not require scaling of the values after smoothing, it is necessary to normalise each value by dividing by the total area under the curve, so that all values add up to 1. Once the $2i+1$ normalised Gaussian values:

$G_{k-i}, G_{k-i+1}, G_{k-i+2}, \ldots\ldots\ldots G_k, G_{k+1}, G_{k+2}, \ldots\ldots\ldots G_{k+i}$, are computed, then each A_j is multiplied by its corresponding Gaussian value and summed to give the new value.

$$A_{ks} = \sum_{j=k-i}^{k+i} G_j\, A_j \quad (3.3.5.2)$$

where G_k is the central and largest multiplier, giving the greatest weight to A_k.

Substituting Eq. 3.3.5.1 in Eq 3.3.52, we get a form which is programmable:

$$A_{ks} = \sum_{j=k-i}^{k+i} exp\left[-\frac{(A_k - Aj)^2}{2(t_k - t_j)^2}\right] A_j \quad (3.3.5.3)$$

3.3.6 Application of the Nyquist Theorem

Hitherto, this theorem has been used overwhelmingly by electrical engineers,[27] acoustic engineers,[28] digital musical instrumental engineers,[29] and astronomers,[30] for improved resolution of noisy analog signals after digitisation.

The Nyquist theorem has as a necessary and sufficient condition applicable to a class of mathematical functions with Fourier transforms which are non-zero inside a finite region of frequencies. When reducing a continuous function to a discrete non-continuous sequence as happens in ADC, the result is non-continuous and therefore not differentiable (or at best only piecewise differentiable) and to regain continuity requires interpolation back to a continuous function. The fidelity of this resulting continuous function depends on the high enough data density or DAQ of the original sample such that no significant information is lost in the sampling process.

So, for ADC to result in faithful reproduction of the original analog signal, sample slices of the analog waveform must be taken frequently, and this means that a high DAQ is well suited to this form of analysis.

The Nyquist Theorem leads to a formula for increasing precision in reconstructing the original continuous-time function from the sample. Reconstruction is possible if high DAQ is not present, provided key absorbance conditions are known. In cases when high DAQ is not satisfied, utilizing key absorbance conditions allows for approximate reconstructions. The fidelity of these reconstructions can be verified and quantified utilizing Bochner's theorem.[31]

Nyquist's result is that data with two or more points per cycle in high DAQ, allows increasing fidelity of reconstruction with increasing DAQ via the Cardinal Theorem of Interpolation.[32]

The computational method used in this discussion to achieve smoothing uses an adaptation of the Nyquist Theorem shown in simple form below. This involves the equation:

$$A_{ks} = \frac{\Psi}{2^{10+n}}\left[\frac{\sum_{j=1}^{4^n} A_j + 2^{n-1}}{2^n}\right] \qquad (3.3.6.1)$$

Where A_{ks} is output proportional absorbance (in mAU);

Ψ is the scaling factor used to convert the absorbance to appropriate units via internal reference of the ADC.

n is the number of additional bits (i.e. number of bits more than 10) of resolution;

A_j are individual analog data input values from the photodiode assembly.

An applied version of equation 3.3.6.1 is shown in the concrete example below:

$$A_{ks} = \frac{1100}{2^{10+5}}\left[\frac{\sum_{j=1}^{4^5} A_j + 2^{5-1}}{2^5}\right] \qquad \text{so}$$

$$A_{ks} = \frac{1100}{2^{15}}\left[\frac{\sum_{j=1}^{1024} A_j + 16}{32}\right]$$ which has adapted the Nyquist Equation 3.3.6.1 as follows:

- $\Psi = 1100$ so this has normalised the absorbance values A_j as an output in mV to the internal reference of 1.10 V in the ADC of the Arduino.
- $n = 5$, which means that the resolution has increased from 10 bits to 15 bits.
- This increased resolution has come about by the oversampling of a data window of $4^5 = 1024$ points.

One consequence of the last point is that the Nyquist Theorem does not require an odd number of data points. This is because increased smoothing is due to the increase in resolution which dictates the window size, and not by an imposed symmetrical smoothing function.

The adaptation of equation 3.3.6.1 allows increased resolution by oversampling, but there are two constraints:

(i) The baud rate must be substantially greater than the maximum sampling rate.

(ii) Any gains in resolution are accompanied in real time by an exponential decrease in DAQ. This is due to the increased value of 4^n in the equation, and the subsequent slowing in real-time processing by the operating system software. The principle is the same as that used in sound systems to obtain ever-improved digital approximation of analog sounds; it is also used in astronomy where many snapshots taken of the same object will give improved resolution when they are overlaid.

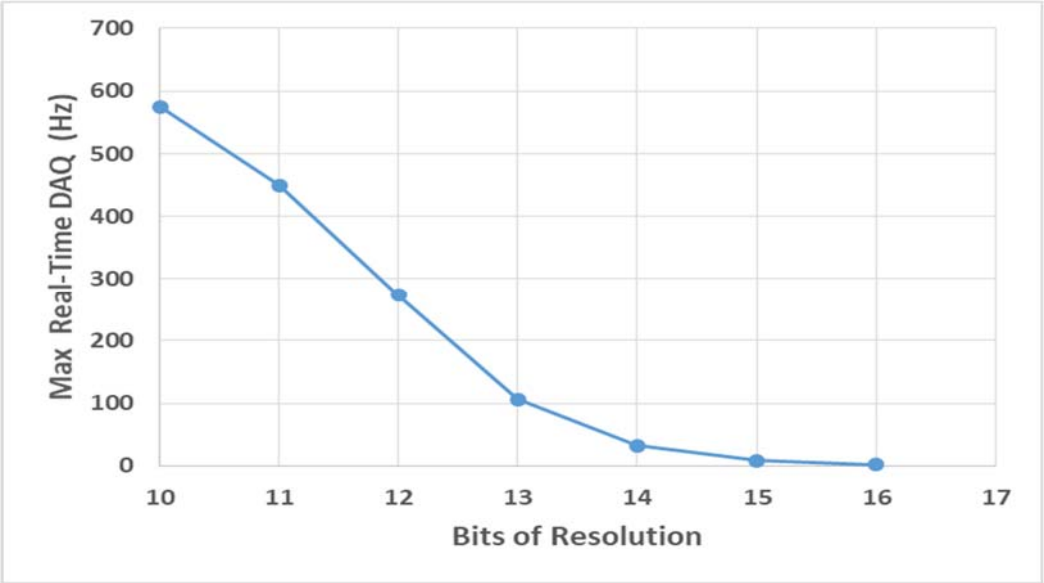


Figure 3.3.6.1: The relationship between DAQ and bits of resolution is shown to be negative exponential. This data is taken from an older model Arduino UNO with 10-bit ADC.

The Nyquist Theorem was derived in order to increase resolution of analog signals; consequently it is permissible to have any value of n. This allows us to treat equation 3.3.6.1 as a continuous function – an approximation which becomes increasingly accurate as core processor speeds (hence baud rate) and DAQ get faster. The DAQ can then be adjusted by alteration in the value of n.

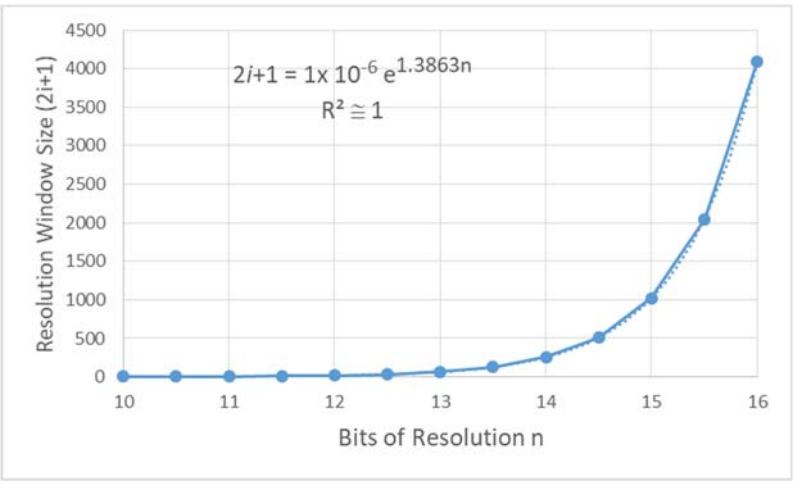


Figure 3.3.6.2: The relationship between oversampling number and bits of resolution for the older model Arduino UNO with 10-bit ADC was determined experimentally, and is shown to be a reliable exponential relationship.

As a concrete example, if $n = 4.84$ then the oversampling window size $2i + 1 = 823$ points by substitution in the equation of the curve in Figure 3.3.6.2. The relationships posited in Figure 3.3.6.1 and Figure 3.3.6.2 were introduced in Chapter 2 §2.2.1 and will be used in further depth again in Chapter 5 §5.2.2.

3.4 Comparative Application of Algorithms to Chromatographic Analysis

To commence a comparative study of smoothing algorithms, it was decided to begin with a simple single peak electropherogram in order to initially identify differences, similarities and possible trends.

An initial single peak electropherogram with DAQ at 1052 Hz was chosen, and the original dataset was compared with two common smoothing algorithms, both at window size of 1023 points. This effectively means each point is smoothed using a cluster of 511 points on each side.

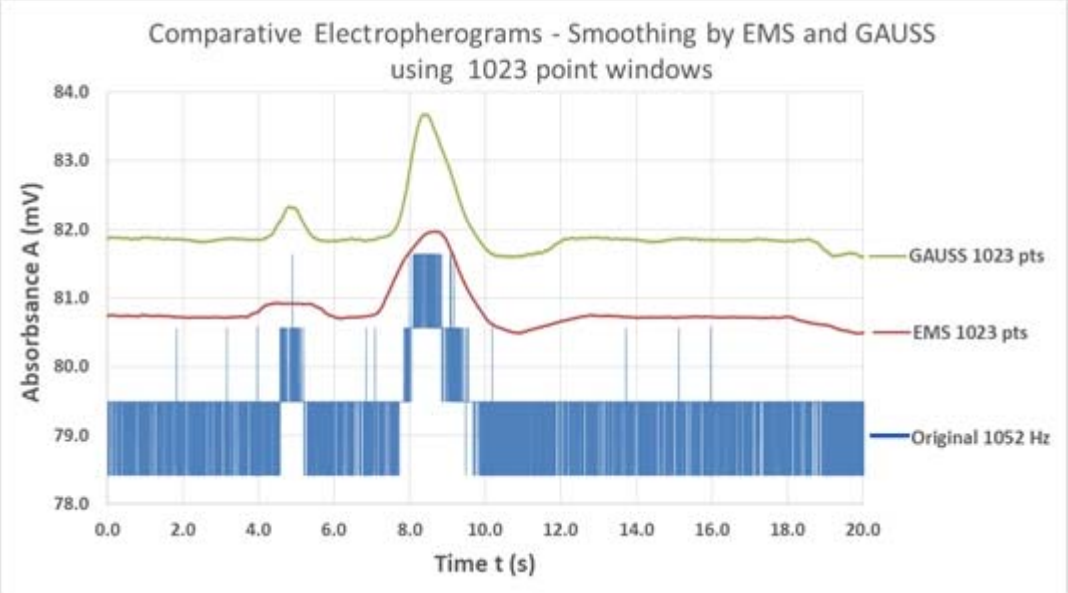


Figure 3.4.1: Dataset for electrophoretic detection of fluorescein with DAQ=1052 Hz, with comparisons using post-facto processing by Exponential Mean Smoothing and Gaussian Smoothing. For this Arduino-controlled system, a fused silica capillary is used with $L_D = 12.5$ cm; $L_T = 15.0$ cm; V= 6.0 kV; E=400.0 V.cm^{-1}; 25 µm i.d. and 363 µm o.d. with standard polyimide coating.

This raised questions about the detection of small peaks, at what appears to be the EOF at around 5.0s in Figure 3.4.1 above, and so it was decided to begin by looking at an

electropherogram which contained an identifiable peak as well as a feature completely obscured by noise. The selected single-peak electropherogram selected was that shown below:

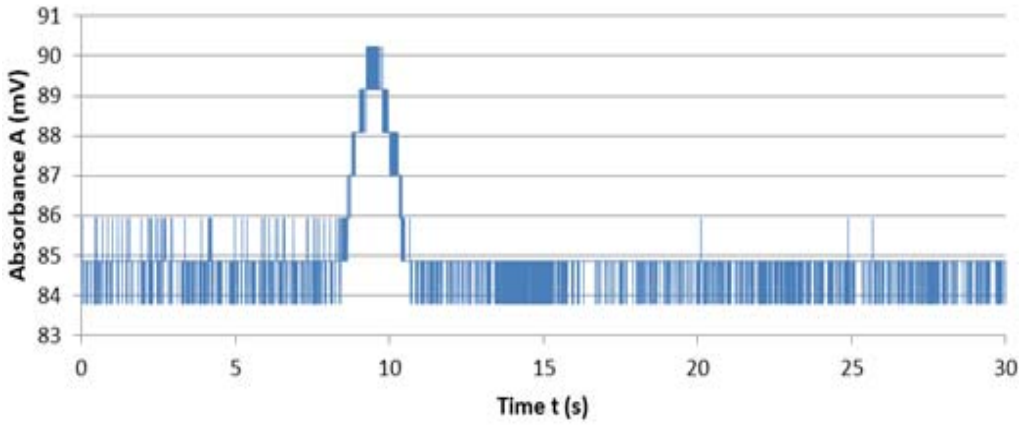


Figure 3.4.2: New dataset for hydrodynamic injection and subsequent detection of fluorescein with DAQ=1052 Hz. For this Arduino-controlled system, a fused silica capillary is used with L_D = 12.5 cm; L_T = 15.0 cm; 25 µm i.d. and 363 µm o.d. with standard polyimide coating. The feature completely obscured by noise lies at around 14.0 s.

A detailed study of this single peak electropherogram was carried out, in which the performance of the Nyquist Theorem was tested against each of the other five common algorithms. In particular, the elucidation of features within the electropherogram which are initially obscured completely by noise is of particular interest.

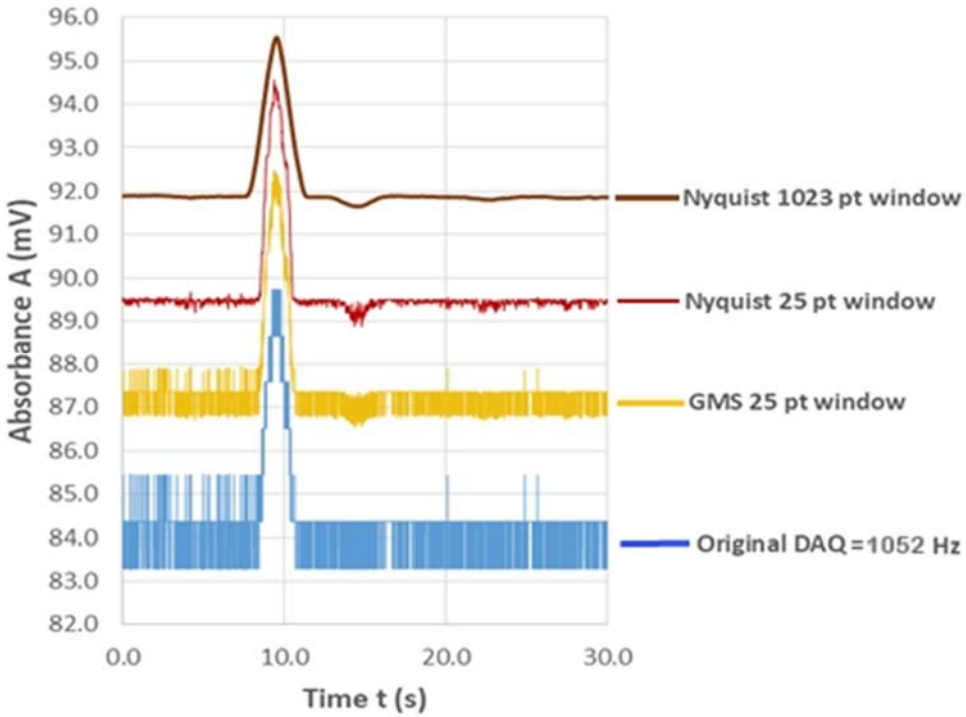


Figure 3.4.3: Dataset for electrophoretic detection of fluorescein with DAQ=1052 Hz, with initial comparisons using post-facto processing by Geometric Mean and Nyquist smoothing. The two Nyquist windows and the GMS window reveal the hidden dip in the baseline featured at around 14.0 s. Further, a visual comparison of Nyquist and GMS with a 25 – point smoothing window suggest that not all algorithms are equally efficient smoothers.

Figure 3.4.3 illustrates that two different smoothing algorithms taken from the set 3.3.1 to 3.3.6 above and applied *post-facto*, begin to reveal some detail at around 14 s -15 s and a much smaller feature at around 25 s after smoothing. Both of these small features are completely obscured in the original low-resolution 10-bit ADC noise of Figure 3.4.2.

On the basis of these and other similar observations in other datasets, it was decided that the following parameters which are critical to chromatographic analysis needed to be addressed:

(i) Peak height;

(ii) Signal/Noise ratio SNR

(iii) Peak width Wp

3.4.1 Initial Comparative Study: Single Peak Electropherogram

The electropherogram described in Figure 3.4.2 was subjected to each of the smoothing algorithms listed in §§3.3.1 to 3.3.6 above, and there were 59 different window sizes which varied from 5 to 1023 points. Peak widths were determined in each case using the vector method outlined in §3.2.3 above. This enabled a precise relationship to be established between peak width and smoothing window size for each of the six algorithms listed in §§3.3.1 to 3.3.6.

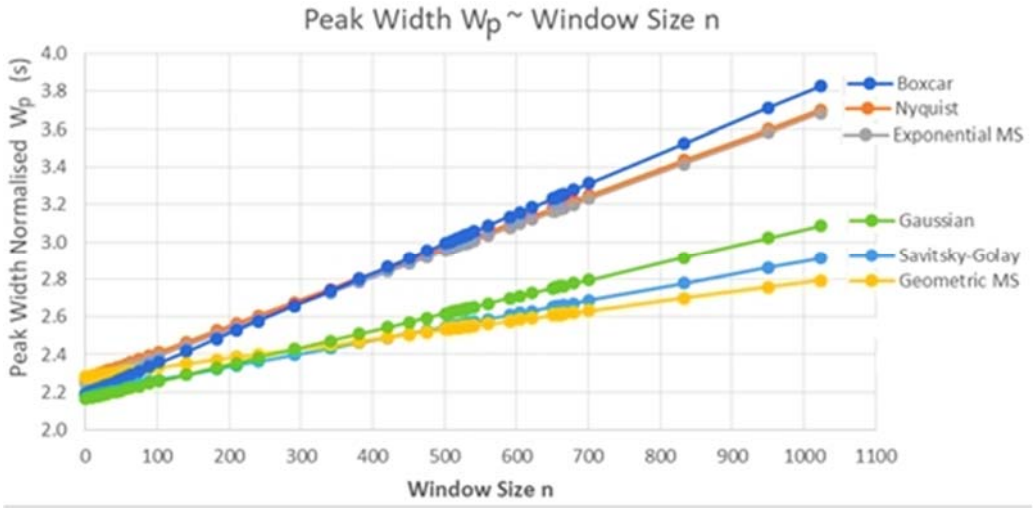


Figure 3.4.1.1: Illustration of the relationship between normalised peak width and the size of the smoothing window for the six algorithms of this study. In all cases, linear approximation of the relationships gives values of $R^2>0.93$,

While the relationships shown in Figure 3.4.1.1 appear at first sight to be linear in all cases, this is not strictly speaking the case. Detailed analysis shows that the only truly linear relationship is Boxcar smoothing which gives the most rapid degree of peak widening. On close analysis by Numerical Methods using log Wp~n or log Wp~log n, all the other relationships begin to show a slowly increasing tendency towards either logarithmic, exponential, or polynomial relationships between the variables at values of $n > 800$. For example, the Savitsky-Golay algorithm begins to show a very gentle (flattening) logarithmic relationship between peak width and smoothing window size when n increases beyond a window size of 700 points.

In fairness, the effects are so slight that window size n would have to be much larger for such relationships to become more obvious at first sight. For the purposes of this discussion at the values of n used here, all the relationships are assumed to be linear.

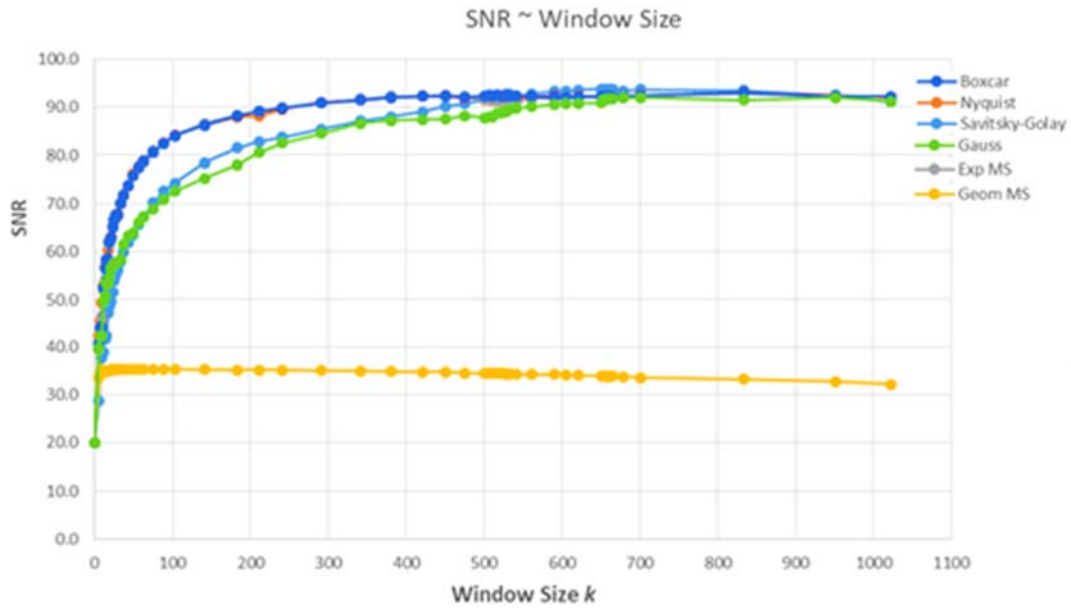


Figure 3.4.1.2: Illustration of the relationship between SNR and the size of the smoothing window for the six algorithms of this study.

In all six cases, the relationships between SNR and smoothing window are logarithmic over the range of window sizes analysed here. It turns out that Nyquist, Boxcar and EMS are difficult to separate. This close relationship means that in order to choose the most efficient algorithms of the six, it becomes apparent that may be necessary to look at variation in more than one criterion simultaneously in order to discriminate more comprehensively. So it was decided to look at peak height rather than SNR to see if a better separation of algorithm performance could be achieved.

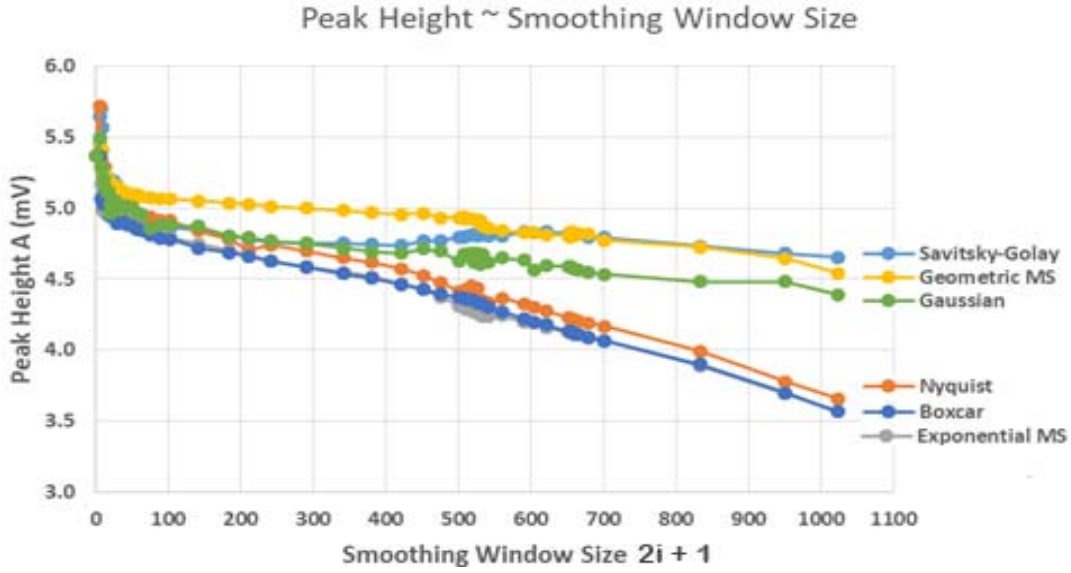


Figure 3.4.1.3: Illustration of the relationship between Peak Height and the size of the smoothing window for the six algorithms of this study.

In Figure 3.4.1.3:

- EMS and Boxcar are almost indistinguishable, and
- peak height~window size of the Nyquist algorithm is almost parallel to that of Boxcar.

This gives a different set of performance outcomes to those shown in Figure 3.4.1.2 for ranking these algorithms, and so these criteria had to be extended to combining pairs of criteria in an effort to discriminate more clearly between smoothing performances of the six algorithms being studied. More specifically, what is needed is to identify the places where SNR is greatest whilst at the same time peak widening is smallest. In order to do this it was necessary to look at the functions of both SNR~n and Wp~n.

By way of explanation, two closely related data curves were constructed which modelled the shapes of each of SNR~n and Wp~n shown in Figures 3.4.1.2 and 3.4.1.3 above. The place of interest is then where the difference between SNR and Wp is the greatest. In order to do this, each of these four function curves had to be normalised and was converted to a percentage. This initial process is illustrated below:

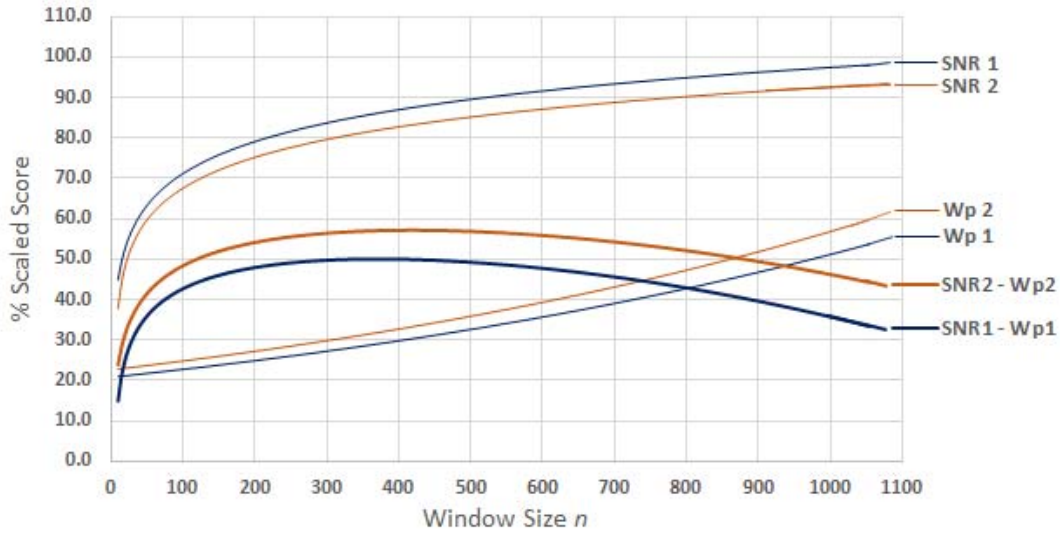


Figure 3.4.1.4: A more clearly-defined separation of algorithmic performance can be seen by the exaggeration of the difference between algorithms 1 and 2, shown as the difference between the endpoints of the functions at SNR1-Wp1 and SNR2-Wp2.

Using this method to enable clearer comparison of different algorithm performance, the SNR and Wp data of this dataset shown in Figure 3.4.3 were both re-scaled to percentages in terms of their respective performance, and a new compound criterion of

the difference between these two newly-scaled criteria (SNR – Wp) showed the more clearly separated set of curves as an indicator of algorithm performance.

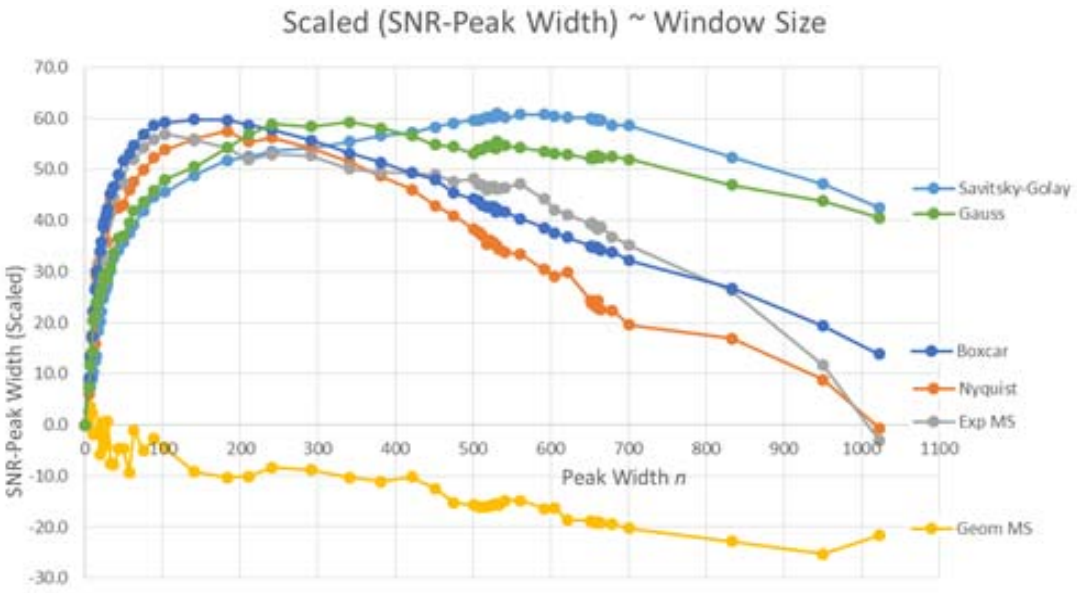


Figure 3.4.1.5: A more clearly-defined separation of algorithmic performance by invoking the compound criterion constructed from scaled data of both SNR and peak width.

Using this functional representation of performance, the relative performance of each algorithm was then quantified in rank order (scaled from 1 to 6) in SNR (with 6 being the best score for highest SNR) and Wp (with 6 being the highest score assigned to the algorithm giving the smallest increase in peak width) to find the best-performing algorithms. These were the algorithms which gave the highest SNR with the smallest signal widening.

Table 3.4.1.1: Performance Comparisons of six algorithms based on scaled SNR and peak width Wp.

Algorithm Window Size (pts)	**SNR**	**Wp**	**TOTAL**	
Savitsky-Golay 605 point	5.9060	6.0000	11.906	Best 3 algorithms for future Composite
Nyquist 241 point	6.0000	5.8453	11.845	
Gauss 381 point	5.7770	5.8012	11.578	
Exponential MS 183 point	5.8800	2.8791	8.759	
Boxcar 211 point	5.8920	1.0000	6.892	Worst 2 algorithms Discarded
Geometric MS 141 point	1.0000	5.8140	6.814	

To summarise the performance of the 6 algorithms, it turns out that Savitsky-Golay and Nyquist have the best performance on SNR whilst also giving the smallest increases in peak width. The best 3 algorithms will be used in chapter 4 for the development of composite smoothing methods.

3.4.2 Comparative Study: Multiple Peak Electropherogram

In the study of the multiple peak electropherogram, it turns out that the analysis of paragraph 3.4.1 above holds good. This means that using the C-102+F+FITC electropherogram of Figure 3.3.2 with DAQ = 875Hz, for all three peaks in the electropherogram the order of performance with respect to SNR and Wp is the same as in Table 3.4.1.1,albeit with slightly different scaled values. It needs to be said however that in the case of peaks 2 and 3, resolution of this pair (which is not relevant in the case of the single peak of Figure 3.3.1) becomes an issue because which affected these values.

The order of performance with regard to resolution, however shows a slight variation, but the top three performers are unchanged as a set.

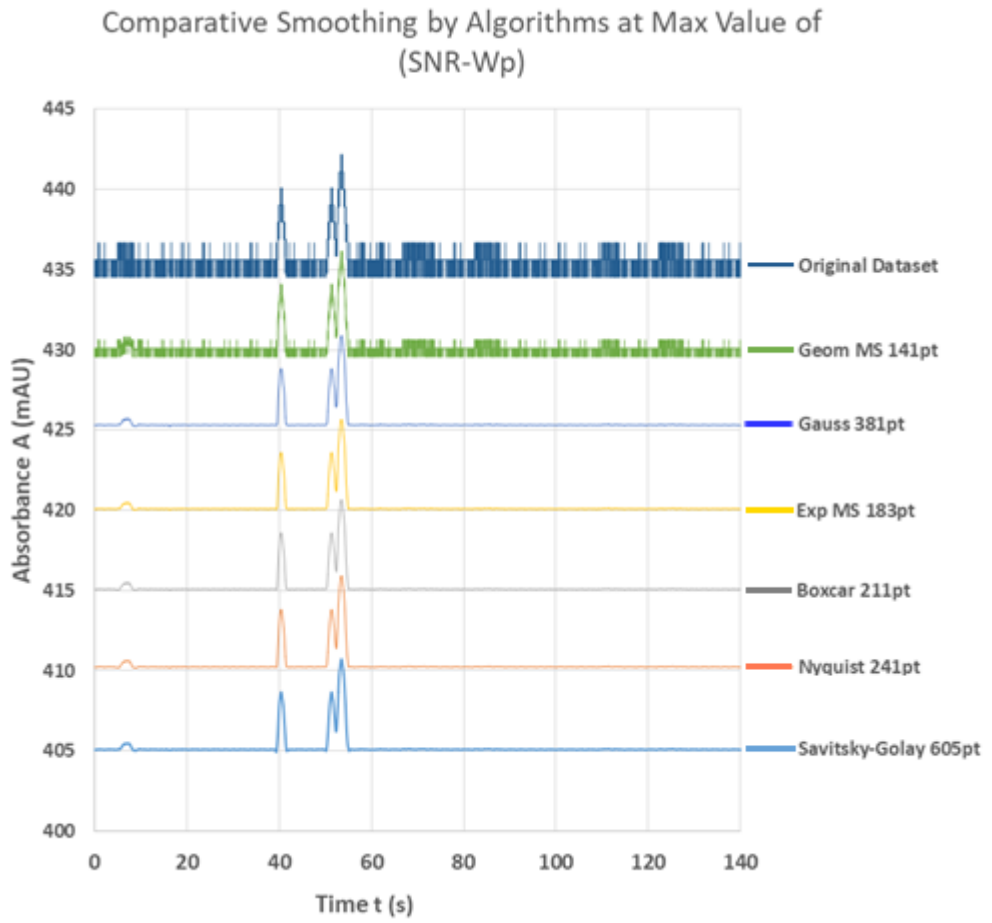


Figure 3.4.2.1: Illustration of the original C-102+F+FITC multi-peak electropherogram together with the six algorithms at the optimum window width for SNR-Wp applied to the largest peak.

For each of the six algorithms, the resolution between peaks two and three at this optimum window width was computed in every case. The algorithms were then ranked according to performance, and re-scaled in accordance with the method used for the single peak electropherogram used in paragraph 3.4.1 above.

Again, the Arduino UNO – based instrument described in Chapter 2 was used for these electropherogram datasets and the *mills()* function was used to ascertain time intervals.

Table 3.4.2.1: Performance comparisons of six algorithms based on resolution, and rank scaled.

Algorithm Window Size (pts)	Resolution	Rank	Scaled Value
Nyquist 241 point	0.89	6	6.000
Savitsky-Golay 605 point	0.84	5	5.373
Geometric MS 141 point	0.82	4	4.795
Exponential MS 183 point	0.78	3	3.669
Gauss 381 point	0.75	2	2.797
Boxcar 211 point	0.66	1	1.000

The anomalous appearance of geometric mean smoothing at ranking number 4 is due to the fact that the optimum performance of GMS at 141 points seems to be because GMS is a very inefficient smoothing method, requiring a much larger window size to achieve the same SNR improvement as any of the other five algorithms.

Boxcar has the poorest performance at the optimum window size for maximum SNR at given Wp because it achieves smoothing very quickly for small window size, but this clumsy statistical method quickly flattens peaks and reduces resolution.

Exponential mean smoothing remains at ranking position 3 as before, Savitsky-Golay and Nyquist are still in the first two positions and Boxcar is still in one of the elimination ranks.

Table 3.4.2.2: Performance comparisons of six algorithms based on all three criteria of: SNR; Peak Width; and Resolution, after rank scaling.

ALGORITHM Window Size (pts)	SNR	Wp	RESOLUTION	TOTAL	RANK
Nyquist 241 point	6.000	5.913	6.000	17.913	6
Savitsky-Golay 605 point	5.893	6.000	5.373	17.266	5
Gauss 381 point	5.811	5.801	2.797	14.409	4
Exponential MS 183 point	5.880	2.894	3.669	12.443	3
Geometrical MS 141 point	1.000	5.794	4.795	11.589	2
Boxcar 211 point	5.817	1.000	1.000	7.817	1

When resolution is added as a third selection criterion, the only difference is that Nyquist and Savitsky-Golay are interchanged when compared to Table 3.4.1.1. As a group, Gauss, Savitsky-Golay, and Nyquist remain as the top three; exponential mean smoothing remains at position 4, and boxcar and GMS remain in the bottom two, although their order is interchanged compared to table 3.4.1.1 - again because of a difference in performance on resolution.

3.5 Conclusions

In this chapter, some consistent mathematical frameworks were established for the ideas involved in signal smoothing which could usefully be extended to enable comparisons between five common smoothing algorithms all of which use a weighted averaging (linked to a statistical, exponential or polynomial function) with the Nyquist Theorem which in unlike any of the others as it smooths a noisy analog signal by increasing the resolution.

The six algorithms developed were then ranked using the criteria of signal-to-noise ratio, peak width and resolution. The aim of such a ranking is to get maximum

smoothing for the smallest window size and hence hopefully the least amount of signal distortion.

The rankings of §3.4.1 and §3.4.2 above can now be used in Chapter 4 to construct and test composite functions to further optimise smoothing.

3.6 References

1. Laude, N. D., Atcherley, C.W., Heien, M.L., Rethinking Data Collection and Signal Processing – Real Time Oversampling Filter for Chemical Measurements. *Anal. Chem.* **2012,** *84* (19), 8422−8426.

2. O'Haver, T. C., An Introduction to Signal Processing in Chemical Measurement. *Journal of Chemlcal Education* **1991,** *66* (6), A147-A150.

3. Wahab, M. F., Patel, D. C., Armstrong, D.W., Peak Shapes and their Measurements - The Need and the Concept behind Total Peak Shape Analysis. *LC•GC Europe* **2017,** (December 2017), 670-678.

4. Dasgupta, P. K., Chromatographic peak resolution using Microsoft Excel Solver - The merit of time shifting input arrays. *J Chromatography A* **2008,** *1213* (1), 50-55.

5. Morawski, R. Z., Spectrophotometric applications of digital signal processing. *Meas. Sci. Technol.* **2006,** *17* (2006), R117-R144.

6. Wahab, M. F., Dasgupta, P. K., Kadjo, A. F., Armstrong, D. W., Sampling frequency, response times and embedded signal filtration in fast, high efficiency liquid chromatography: A tutorial. *Anal Chim Acta* **2016,** *907*, 31-44.

7. Felinger, A., Kilar, A., Boros, B., The myth of data acquisition rate. *Anal Chim Acta* **2015,** *854*, 178-182.

8. Luo, J., Ying, K., He, P., Bai, J., Properties of Savitzky–Golay digital differentiators. *Digital Signal Processing* **2005,** *15* (2), 122-136.

9. Liu, B. F., Sera, Y., Matsubara, N., Otsuka, K., Terabe, S., Signal denoising and baseline correction by discrete wavelet transform for microchip capillary electrophoresis. *Electrophoresis* **2003,** *24* (18), 3260-3265.

10. Duong, H. A., Le, M. D., Mai-Nguyen, K. D., Hauser, P. C., Pham, H. V., Mai, T. D., In-house-made capillary electrophoresis instruments coupled with contactless conductivity detection as a simple and inexpensive solution for water analysis: a case study in Vietnam. *Environ Sci Process Impacts* **2015,** *17* (11), 1941-1951.

11. Castro, E. R., Manz, A., Present state of microchip electrophoresis: state of the art and routine applications. *J Chromatogr A* **2015,** *1382*, 66-85.

12. Weissman, M. B., Inverse frequency noise and other slow, non-exponential kinetics in condensed matter. *Reviews of Modern Physics* **1988,** *60* (2), 537 - 571.

13. Salmasi, M., Buttner, U., Glasauer, S., Fractal dimension analysis for spike detection in low SNR extracellular signals. *J Neural Eng* **2016,** *13* (3), 1-19.

14. Kakkar, A., Rodrigo Navarro, J., Schatz, R., Pang, X., Ozolins, O., Udalcovs, A., Louchet, H., Popov, S., Jacobsen, G., Laser Frequency Noise in Coherent Optical Systems: Spectral Regimes and Impairments. *Sci Rep* **2017,** *7* (844), 1-10.

15. Shinagawa, M., Akazawa, Y., Wakimoto, T., Jitter Analysis of High-speed Sampling Systems. *IEEE Journal of Solid-State Circuits* **1990,** *25* (1), 220 - 224.

16. Chen, X., Sobhy, E.A., Yu, Z., Hoyos, S., Silva-Martinez, J., Palermo, S., Sadler, B.M., A Sub-Nyquist Rate Compressive Sensing Data Acquisition Front-End. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems* **2012,** *2* (3), 542-550.

17. Santoso, S., Powers, E.J., Grady, W.M., Parsons, A.C., Power Quality Disturbance Waveform Recognition Using Wavelet-Based Neural Classifier. *IEEE Transactions on Power Delivery* **2000,** *15* (1), 222-228.

18. Levant, A., Robust Exact Differentiation via Sliding Mode Technique. *Automatica* **1998,** *34* (3), 379-384.

19. Archibald, B. C., Parameter space of the Holt-Winters' model. *International Journal of Forecasting* **1990,** *6*, 199-209.

20. de Levie, R., Estimating Parameter Precision in Nonlinear Least Squares with Excel's Solver. *Journal of Chemical Education* **1999,** *76* (11), 1594-1598.

21. Chatfield, C., Koehler, A.B., Ord, J.K., Snyder, R.D., A New Look at Models for Exponential Smoothing. *Journal of the Royal Statistical Society* **2001,** *50* (2), 147-159.

22. Halmer, D., von Basum, G.. Hering, P., Mürtz, M., Fast exponential fitting algorithm for real-time instrumental use. *Review of Scientific Instruments* **2004,** *75* (6), 2187-2191.

23. Gijbels, I., Pope, A., Wand, M.P. , Exponential Smoothing via Kernel Regression. *Journal of the Royal Statistical Society* **1999,** *Series B* (61), 39-50.

24. Gorry, P. A., General Least-Squares Smoothing and Differentiation by the Savitzky-Golay Method. *Anal. Chem.* **1990,** *62* (12), 570-573.

25. Savitsky, A., Golay, M.J.E., Smoothing and Differentiation of Data by Simplified Least Squares Procedure. *Anal. Chem* **1964,** *36* (8), 1627-1639.

26. Madden, H. F., Comments on the Savitzky-Golay Convolution Method for Least-Squares Fit Smoothing and Differentiation of Digital Data. *Anal. Chem* **1978,** *50* (9), 1383-1386.

27. Ionescu, M. V., Sato, M., Thomsen, B. C., In-band OSNR Estimation for Nyquist WDM Superchannels. In *Proceedings of the European Conference on Optical Communications (ECOC) 2014, Cannes* 2014; pp P.4.16.1 - P.4.16.3.

28. Breithaupt, C., Gerkmann, T., Martin, R., Cepstral Smoothing of Spectral Filter Gains for Speech Enhancement Without Musical Noise. *IEEE Signal Processing Letters* **2007,** *14* (12), 1036-1039.

29. Emiya, V., Badeau, R., David, B., Multipitch Estimation of Piano Sounds Using a New Probabilistic Spectral Smoothness Principle. *IEEE Transactions on Audio, Speech, and Language Pprocessing* **2010,** *18* (6), 1643-1654.

30. MacMahon, D. H. E., Price, D. C., Lebofsky, M., Siemion, A.P. V., Croft, S., The Breakthrough Listen Search for Intelligent Life: A Wideband Data Recorder System for the Robert C. Byrd Green Bank Telescope. *Publications of the Astronomical Society of the Pacific* **2018,** *130* (986), 1-19.

31. Nemirovsky, J., Shimron, E., Utilising Bochner's Theorem for Constrained Evaluation of Missing Fourier Data. *Israel Institute of Technology, Haifa* **2015**, 1-18.

32. Candes, E. J., Romberg, J., Tao, T., Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on Information Theory* **2006,** *52* (2), 489-509.

4 Construction of Composite Algorithms: A Comparative Study of Smoothing and Resolution

4.1 Introduction: Conditions for Simultaneous Algorithms

In the *post-facto* application of the Savitsky-Golay method for smoothing described in Chapter 3, it was decided to use a 5-point polynomial (polynomial degree $= 4$) because in real time, a program which additionally uses a compound of two algorithms and processes data *in situ* is significantly slowed by any polynomial of degree >4. The process used is outlined in Figure 3.3.4.1 in Chapter 3.

From this strategy arose the notion of nested algorithms, and the mathematics to extend this was then developed to underpin this idea. This is not an adaptation of successive application of algorithms, nor is it implicitly a simple iterative process but a computational nesting which is iteratively applied along the entire length of n data points of the signal in a two-dimensional manner which takes account of both ordinate and abcissal dimensions.

Before beginning the mathematical approach to the computational structure of nesting, it is firstly necessary to clarify some foundations:

a) There is an implicit assumption in every smoothing algorithm that the signal shape which needs to be revealed or clarified is of the form pre-determined by the algorithm. This means that within a given dataset, an algorithm is pre-programmed to search for its own pattern.[1]

b) Every dataset is a set of points which are inherently discrete, and so the dataset is also discontinuous. The advantage of very rapid DAQ is that discontinuities are very small and so can be well approximated by a continuous function. This is an important condition because without a continuous function many of the tools for analysis such as differentiation and homogeneity are lost.

c) The result of an assumed underlying smoothing algorithm is that the greater the window size, the more strongly the underlying assumed pattern begins to be imposed upon the dataset, and peaks begin to assume the shape of the imposed smoothing algorithm. This is illustrated in Fig. 3.2.1.1, where the Gaussian shape and an Exponential decay become apparent at window size of 1023 points when applied to the electropherogram. Since the noise which is to be

removed is also assumed to be random, it is excluded because randomness fits no underlying patterns.

4.1.1 Three Methods: Convolution of Algorithms; Iterated and Recursive Algorithms; and Compound (Nested) Algorithms

Before embarking on the experimental section of this chapter, it is instructive to look at three different methods of using multiple algorithms for signal smoothing. Some of these methods come from astronomy, general image processing, Aeronautics and radar, and in general from disciplines in which a desired signal or image needs to be unscrambled from noise, clutter or other distractors.[2]

This short discourse on three methods of using two algorithms will hopefully clarify why the third method was selected in preference to the other two.

4.1.1.1 Convolution of two functions

Convolution of two functions f and g (denoted $f*g$) is a mathematical strategy which produces a third function, $h = f*g$, which is actually a version of f which has been modified by g.

The convolution process involves integration of the pointwise multiplication of f and g, as a function of the amount by which f is pointwise translated. Convolution has applications that include image and signal processing, and has been applied in chromatography and instrumental signal processing in chemical analysis. Schnöll-Bitai explains the use of this method by convoluting a Gaussian algorithm with a version of EMS to improve SNR and preserve signal characteristics.[3]

Schnöll-Bitai shows that convolution of two algorithms with the same mathematical structure (such as Gauss and EMS which are both exponential in form) does show a modest increase in SNR, but in terms of real-time pointwise processing of fast DAQ, it is computationally expensive in terms of time needed to execute convolution in real time during signal acquisition. The method also raises a problem of rapid peak broadening, and hence a widening of the distance between the turning points (t_{left}, A_{left}) and

(t_{right} , A_{right}). This leads to the added complexity of trying to determine a correction algorithm (e.g. by using a statistical bivariate deconvolution within each time window and for each value of $2i+1$) to calculate the relationship between the size of smoothing window $2i+1$, and the associated introduced error.[4]

Since all the chapters above deal with fast data acquisition, it becomes necessary to briefly address Fast Fourier Transform (FFT) as a smoothing method; it has been used as such for over 25 years,[5] and it is important to show why it is not suitable for inclusion in this particular study:

a) Convolution by using Fast Fourier Transform (FFT) may mitigate both time delay and error correction to varying degrees, but effectively the smoothing of the same noise using two functions of related mathematical structure (sine and cosine in the case of a Fourier Series) means that a limit to the benefits of smoothing is quickly reached in terms of window size. Because of this similarity, increasing window size past some optimal value as illustrated in Figure 3.4.1.3, shows that additional smoothing has negligible effect on improvement in SNR. This is because with increasing window size, the disadvantage of peak widening causes peak height reduction to a degree greater than the advantages of noise reduction, and noise reduction quickly reaches a limit due to the exponential decay structure of two convoluted functions.

b) Fast Fourier Transforms are well suited to processing of stationary signals; in such signals the time domain is constant. However for moving signals of variable heights, shapes and widths it becomes inefficient because information is lost in the time domain.[6] This deficiency is sometimes mitigated by introducing a time window that represents a compromise between the time and frequency.

c) Direct application of a Fourier Transform (FT) of large data sets (such as those used in Chapter 5 to follow) involves an unmanageable number of computations. This has been known for several decades and so the Cooley-Tukey algorithm is a shortcut which forms the basis for a Fast Fourier Transform, which renders computation more manageable.[7] The Fast Fourier Transform is then already separated from the raw data by one embedded algorithm.

For these reasons, Fourier transforms are not considered in this study but their current ubiquity in the treatment of signals and noise will form the basis for some future work.

4.1.1.2 Iterative and Recursive Application of two Algorithms

In computational mathematics, an iterative method is a mathematical procedure that uses a looping control structure (for example while, do while, for) in order to repeat a section of code until a certain condition is met.

Some concrete examples of such conditions[8] in the case of an electropherogram could be:

- "when σ_{At} decreases to below a specified value", or
- "when SNR increases to above a pre-determined value".
- "when the value of $\frac{\sigma_{At(new)}}{\sigma_{At(original)}} < 0.1$", or its equivalent;
- "when $\sigma_{At(j-1)} - \sigma_{At(j)} < 0.01$."

An iterative method such as in the last two examples described above, are *convergent* because the corresponding sequence of σ_{At} values converges until $\frac{\sigma_{At(new)}}{\sigma_{At(original)}} < 0.1$.

In setting such an end condition, it is necessary to be realistic in terms of computing time, because an asymptotic outcome does not become more functionally useful beyond a certain number of iterations as shown in Figure 4.1.1.2.1 below. In that figure, iterative smoothing windows varying in size from $2i+1 = 9$ to $2i+1 = 1023$ points were used on separate computational spreadsheets, each of which determined all the relevant data calculations of significance, including σ_{At}.

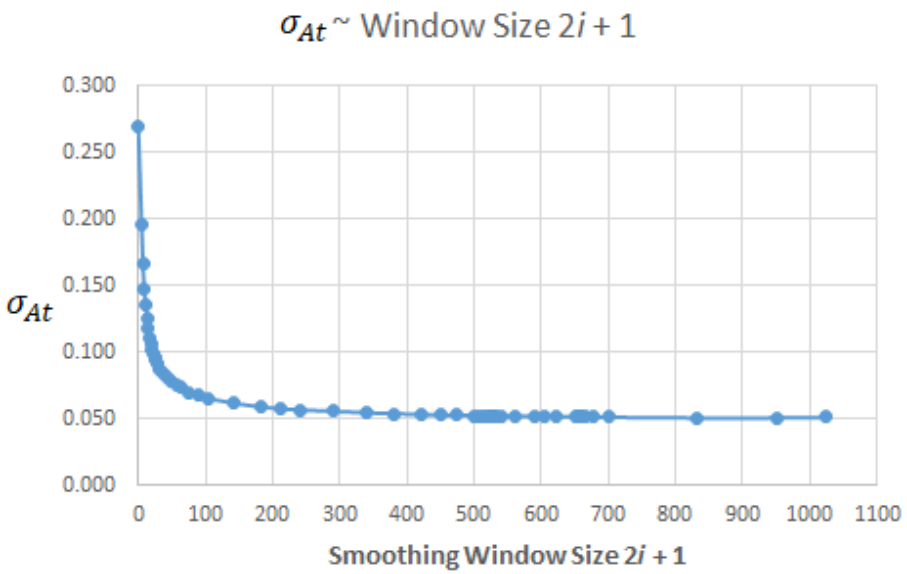


Figure 4.1.1.2.1: Decrease in two-dimensional standard deviation (noise) on smoothing by Savitsky-Golay algorithm when applied to the original dataset for electrophoretic separation of fluorescein, fluorescein isothiocyanate and coumerin-102 shown in Figure 3.3.2. DAQ = 875Hz and total time of the dataset = 140 s. This is an excerpt from a dataset in which smoothing window $2i + 1$ is iterated from $i = 4$ to $i = 511$.

The data points in Figure 4.1.1.2.1 were obtained using an adapted and simplified convergence method applied to the dataset and adapted from that proposed by Doucet and Andrieu.[9] A mathematically rigorous convergence analysis of an iterative method would need to be performed beforehand on any potential pair of nested algorithms used in this way to estimate the effect of changes in variables on each iteration.

Iterative Algorithms have the advantage of fast performance; however, it seems as if iterative methods could be useful for post-facto analysis in very large numbers of data points from high DAQ.

Recursion is when an algorithm repeats itself in a program ("calls" itself in a software loop) until a predetermined condition is met. Recursive algorithms must obey three conditions:

(i) A recursive algorithm must have a base case (a condition);

(ii) A recursive algorithm must change its state *and* move toward the condition – i.e. two software checks on each recursion;

(iii) A recursive algorithm must call itself recursively in the smoothing program.

To obey the second law, we must arrange for a change of state on each recursion that moves the algorithm toward the base case. A change of state means that some data that the algorithm is using is modified in order to move towards the base case (initial precondition).

A concrete example applied recursively to raw data from an electropherogram might look like this:

1. Initial precondition: $\sigma_{At} < 0.1$;
2. Run Savitsky-Golay smoothing with 5-point polynomial and 13-point window;
3. Calculate σ_{At};
4. Check size of (σ_{At} - 0.1). If the value is negative, then STOP. Else go on;
5. Depending on the result of Step 4 above, increase polynomial to 7-point and the size of smoothing window $2i+1$ by an amount related to the size of $|\sigma_{At} - 0.1|$, i.e. the distance between the target value of 0.1 in the base case and the last value of σ_{At};
6. Go to Step 3 with new polynomial and window size and repeat until precondition (base case) at Step 1 is met.

In smoothing of electropherograms, this would be of little value if the same algorithm was repeated without Step 5 – it is already established above that such a process increases error due to peak widening and leads to additional complexity in terms of error correction. Current optimised algorithms for CE data processing such as de-noising, peak detection, and migration time alignment will still include arbitrary parameters that need to be optimized by the data analysts.[10] Recursion of two algorithms of different form in succession has the same outcome; it is as if one recursion is followed by another and then the process is repeated until the conditions are met. This is still the equivalent of a single algorithm being repeated.

Advantages of recursive methods are that

- they can be applied backwards;[11] in such cases, algorithmically appealing recursions can be used to calculate the smoothing density;

- they can be applied in conjunction with filtering techniques; such recursion shows that once we have computed (or reliably estimated) smoothing densities over the entire dataset, then it is possible to execute a backward recursion as in the forward filtering-backward smoothing recursion presented by Kitagawa.[12]

- they can be used to determine a critical parameter by numerical methods implicit in a C++ or similar program; for example the determination of a maximum SNR in a signal as shown in Figure 4.1.1.2.2 below:

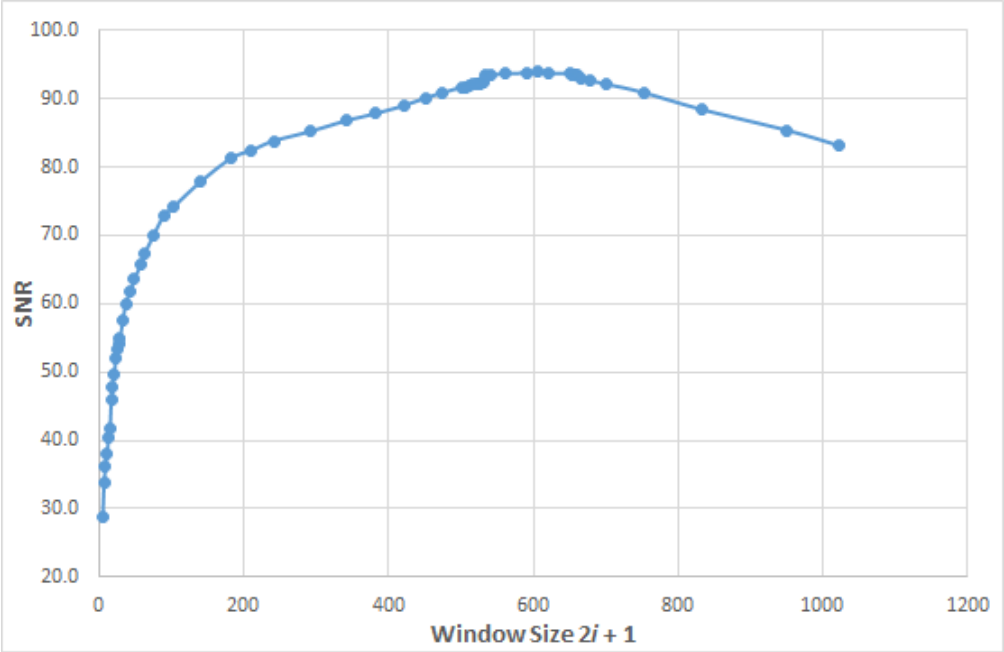


Figure 4.1.1.2.2: The change in SNR on smoothing by 5-point Savitsky-Golay algorithm when applied to the largest signal of the original dataset for electrophoretic separation in Figure 3.3.2. DAQ = 875Hz and total time of the dataset = 140 s. The smoothing window $2i + 1$ is iterated from $i = 4$ to $i = 511$ i.e. window size 9 to 1023 points.

As soon as the value of $\frac{SNR_2 - SNR_1}{window_2 - window_1} < 0$, then the value of SNR_2 retained from the previous recursion is the desired maximum value of SNR. In the case of the example shown in Figure 4.1.1.2.2 above, this value is at $i = 302$ i.e. window size of 605 points.

In terms of programming *in situ* during data acquisition, recursive functions have to keep the function records in memory and jump from one memory address to another to invoke the next recursion, self-test to pass one or more parameters and return values, and then calculate and readjust parameters for the next recursion. That makes them very time-hungry and expensive in terms of performance,[13] often leading to short-cuts in programming and subsequent need for error correction and estimation.[14]

4.1.1.3 Compound (Nested) Algorithms

When moving to this new level of complexity, it is instructive to begin with two definitions for clarification.

Definition 1: **A Primitive Algorithm** is a set of mathematical and consequent computational rules, which give unambiguous specification on how to solve a class of problems. Algorithms can perform calculation, data processing, automated reasoning tasks, or any combinations of these.

For example, in order to solve a class of problems where noise obscures a signal, an algorithm based on the mathematics of Gaussian approximation can be used to enhance the signal by minimising the noise. This Gaussian approximation is translated into a computer program which acts upon the data to execute the algorithm and solve the problem.

Definition 2: **A Compound Algorithm** is an algorithm in which one or more of the computable steps within the algorithm is a call to execute another (secondary) algorithm.

In this Chapter, the construction of compound algorithms is applied to datasets in order to compare the performance of compound algorithms on the key criteria of SNR, Peak Width and Resolution with the performance of the three primitive algorithms selected in Chapter 3 on the same criteria.

Any algorithm is either a primitive algorithm or a compound algorithm.

The technique of this chapter will be to use the compound function $f \circ g$ or $f(g(t))$ where f and g are functions of different underlying mathematical structure, applied as algorithms to the same dataset at the same time, and signal clarification is dealt with by two simultaneous but different underlying functional models, a mathematical process known as *mollification*.[15]

Before dealing with the more complex issue of nesting and mollification, it must be necessary to clarify each of the three primitive algorithms concerned, and pick out the underlying structures inherent in each. These will be the best three algorithms, selected according to the criteria outlined in Chapter 3. They are given again here for clarification of the form of their underlying mathematical structure and also for completeness, in the following equations and summary:

Savitsky-Golay (605 point window; $i = 302; m = 3$) Polynomial Function

Each of the p points in the cluster are treated with a set of integer convolution coefficients,[16] C_j, where A_m is the centre of the cluster, and $p = 2m - 1$

$$S(t) = \left(\frac{1}{\sum_{j=1}^{p} C_j}\right)\left[C_m A_m + \sum_{j=1}^{m-1} C_j\left(A_{m-j} + A_{m+j}\right)\right]$$

and from the table of convolution coefficients, when the polynomial is of degree 4, then $m = 3$, $p = 5$ and

$$A_3 = \frac{1}{35}[17A_3 + 12(A_2 + A_4) + (-3)\,(A_1 + A_5)] \tag{4.1.1.3.1}$$

$h = \sum_{j=1}^{5} C_j = 35$ is the *integer normalisation constant*.

Nyquist (241 point window; $i = 120$) Power Function

$$A_{ks} = \frac{\Psi}{2^{10+n}}\left[\frac{\sum_{j=1}^{4^n} A_j + 2^{n-1}}{2^n}\right]$$

So $4^n = 241$ and hence $n = 3.956445$. The specific Nyquist algorithm which gives the best results in Chapter 3 is then

$$N(t) = \frac{1100}{2^{13.9564}}\left[\frac{\sum_{j=1}^{241} A_j + 7.7621}{15.52417}\right] \tag{4.1.1.3.2}$$

GAUSS (381 point window) Statistical (exponential) Function

Once the normalised Gaussian values:

$G_{k-i}, G_{k-i+1}, G_{k-i+2}, \ldots\ldots, G_k, G_{k+1}, G_{k+2}, \ldots\ldots, G_{k+i}$,are computed, then each A_i is multiplied by its corresponding Gaussian value and summed to give the new value.

$$A_{ks} = \sum_{j=k-i}^{k+i} G_j\, A_j \tag{4.1.1.3.3}$$

where G_k is the central (largest) multiplier, giving greatest weight to A_k

With three algorithms available, the immediate question is to determine which combination of 2 algorithms will give an optimal outcome for any parameter of interest.

For example:

- Which combination of algorithms gives the best outcome for resolution and peak width? or;
- Which combination gives the best outcome for SNR only?

Table 4.1.1.3: Performance Comparisons of best three algorithms based on scaled scores in SNR, Peak Width and Resolution from Chapter 3. Exponential Mean Smoothing is included for reference only, as explained below.

Algorithm	SNR	Wp	Res	TOTAL	
Nyquist 241 pt	6.000	5.845	6.000	17.845	Best 3 algorithms used for Composite Smoothing functions
Savitsky-Golay 605 pt	5.906	6.000	4.913	16.819	
Gauss 381 pt	5.777	5.801	2.956	14.534	
Exponential MS 183 pt	5.880	2.879	3.608	12.367	Eliminated

The summary in this table enables an answer to such questions because the simple analysis suggests that one would pick the algorithms with the highest scores in each category of interest. A complete table in APPENDIX 4 however shows that on each category, the list changes slightly in order. For example, although Geometric Mean Smoothing ends up being at the bottom of the list of performance scores, for small optimal window size of 141 points, it performs better than Gauss in resolution. This leaves open the possibility of substituting GMS for Gauss in a combination requiring improved resolution only.

4.2 Choice of Algorithm Combination and Order

In Chapter 3, the six primitive smoothing algorithms were applied to the F-FITC-C102 multiple-peak dataset, firstly in order to test their consistency in performance when compared to the single peak dataset with regard to:

a) Signal-to-noise ratio SNR;
b) Peak Width Wp; and then to obtain an additional performance measure in
c) the resolution of closely neighbouring signals.

When they were tested with the F-FITC-C102 dataset, and their performance shown in APPENDIX 4, the following became apparent:

(i) The top three algorithms remained in the top group of three, although the relative performance of each changed position as each of the three criteria are scaled and added.
(ii) The Exponential Mean Smoothing remained unchanged in fourth place no matter which criteria or combination of criteria are used
(iii) The Boxcar and Geometric Mean Smoothing algorithms remained in the bottom two places, on SNR, Wp, and total performance score, and so were excluded for the purposes of this discussion.

Further, when the additional criterion of Resolution was added and scaled, then as noted above, GMS performed better than Gauss.

As shown above, no advantage is to be gained by repeating an algorithm within itself. To clarify, smoothing an electropherogram with Savitsky-Golay using a 603 point window nested into a 605 point window of the same function is to reuse the same method inside itself. The polynomial has already been applied in a best fit manner on the original electropherogram, and a nested application leads to no useful change in the smoothing fit, but only results in peak widening and subsequent decrease in SNR.

4.2.1 Rules for Embedding of Optimised Functions

The data set can be thought of as a string of ordered numbers, the ordering and spacing determined by the time axis and DAQ. To perform the same outcome i.e. smoothing of noise by using two different functions of different form and two different windows at the same time presents a new level of complexity.

The smoothing functions used to construct the new composite functions are each individually dependent (under conditions which are both necessary and sufficient) on window size to optimise all three of SNR, peak width and resolution. When the composite functions are derived, necessary degrees of freedom require that there are four rules which must apply:

Rule 1: An optimal algorithm with a smaller window can only be embedded within an algorithm with a larger window, and not the other way round.

Rule 2: No algorithm can nest inside itself.

Rule3: The algorithm with the smaller window must be iterative, and must iterate within the algorithm with the larger window.

Rule 4: The algorithm with the larger window may be either iterative or recursive.

The result of these 4 rules, and their consequential reduction of degrees of freedom means that there are only three possible configurations of nested algorithms to be considered in this study.

Table 4.2.1.1: Selection of algorithm pairs from the group of three top-performing algorithms, to be used for composite analysis, based on function type, window size and Rules 1-4 above. The outer function is always the larger window size.

FORMAT: **Outer(Inner(t))**	**S-G (605 pt)** **Polynomial**	**NYQ (241 pt)** **Power**	**GAUSS (381 pt)** **Exponential**
S-G (605 pt) **Polynomial**	N	S(N(t))	S(G(t))
NYQ (241 pt) **Power**	S(N(t)) Done	N	G(N(t))
GAUSS (381 pt) **Exponential**	S(G(t)) Done	G(N(t)) Done	N

4.2.2 The Conceptual Shape of Three Pairs of Nested Algorithms

In this study, both the inner and outer algorithms are iterative.[17] Although the outer algorithm may be recursive, that configuration is discussed in Chapter 6 as a possible future direction of interest.

Now, using the framework established in §3.1.1.2:

$$i, j, k, m, n, p \in \mathbb{Z}^+$$

and using the notation introduced for the Savitsky-Golay process of §3.3.4, consider:

Outer algorithm Ξ with window width $2i + 1$, centred at $k = i + 1$ and;

Inner algorithm Θ with window width $2p + 1$, centred at $m = p + 1$, and

$$2i + 1 \geq 2p + 3 \qquad (4.2.2.1)$$

This is so that there is at least one point between the outer and inner windows.

$$(t_{ks}, A_{ks}) = \Xi \{(t_j, A_j) | j = k\text{-}i, k\text{-}i+1, k\text{-}i+2, \ldots, \mathbf{k}, k+1, k+2, \ldots, k+i;\ i, j, k \in \mathbb{Z}^+;\ t_k, A_k \in \Re\} \qquad (4.2.2.2)$$

$$(t_{ks}, A_{ks}) = \Theta \{(t_j, A_j) | j = m\text{-}p, m\text{-}p+1, m\text{-}p+2, \ldots, \mathbf{m}, m+1, m+2, \ldots, m+p;\ j,p,m \in \mathbb{Z}^+;\ t_k, A_k \in \Re\} \qquad (4.2.2.3)$$

The centre point of Θ moves from one end of the Ξ window to the other, so given the constrains of equations 4.2.2.1 – 4.2.2.3 above,

(t_{ks}, A_{ks}) by nesting from Rules 1-4, becomes:

$$(t_{ks}, A_{ks}) = {}^{k+i}_{k-i}\Xi \left({}^{m+p}_{m-p}\Theta\right) \qquad (4.2.2.4)$$

Where both Ξ and Θ are iterative operators.

4.2.3 Programmable Combinations

While equation 4.2.2.4 above is a conceptual statement in which the operators Ξ and Θ are defined by different algorithms, it is not able to answer just how specifically it is to be applied or programmed in the course of data acquisition. What follows is a concrete explanation of how each of the allowed combinations from table 4.2.1.1 above can be written in order for them to be translatable into computer code – either for *in situ* or *post-facto* processing.

4.2.3.1 Combination 1: Nyquist 241 within Gauss 381

The mathematics for coding the algorithm rests on beginning with the Nyquist algorithm iterating through the smaller (241 point) window:

$$(t_{ks}, A_{ks}) = \frac{\Psi}{2^{10+n}}\left[\frac{\sum_{j=1}^{4^n}(t_j, A_j) + 2^{n-1}}{2^n}\right]$$

Where $4^n = 241$, whence $n = 3.953$, so

$$(t_{ms}, A_{ms}) = \frac{\Psi}{2^{13.9564}}\left[\frac{\sum_{j=m-120}^{m+120}(t_j, A_j) + 2^{2.9564}}{2^{3.9564}}\right] \tag{4.2.3.1.1}$$

To be embedded within the Gauss algorithm iterating through a 381 point window:

$$(t_{ks}, A_{ks}) = \sum_{j=k-190}^{k+190} G_j\,(t_j, A_j) \tag{4.2.3.1.2}$$

In this form, the Gauss algorithm is centred at the kth (central) point. If we centre it at the 191st point, it tends to Gaussian symmetry about that point. The values of the kernel G_j are calculated from equation 3.3.5.1 which is repeated here:

$$G_j(A_k, A_j) = exp\left[-\frac{(A_k - A_j)^2}{2(t_k - t_j)^2}\right] \tag{3.3.5.1}$$

With 191 points in a Gaussian distribution, effectively all points beyond $\pm 8\sigma_{At}$ are of negligible influence because $G_j \to 0$ as j increases.

So the iterative composite combining equations 4.2.3.1.1 and 4.2.3.1.2 under the constraints of 4.2.2.4:

$$(t_{ks}, A_{ks}) = \sum_{j=k-190}^{k+190} \frac{G_j \Psi}{2^{13.9564}} \left[\frac{\sum_{j=m-120}^{m+120} (t_j, A_j) + 2^{2.9564}}{2^{3.9564}} \right]$$

Given that Ψ is a scaling constant, and the 191 values of G_j are also constants which need only be calculated once in the whole program (before the first iteration) and subsequently read from memory on each successive cycle; and also combining constants, the equation above simplifies to:

$$(t_{ks}, A_{ks}) = \sum_{j=k-190}^{k+190} G_j \Psi \left[\frac{\sum_{j=m-120}^{m+120} (t_j, A_j) + 7.7621}{2^{17.9129}} \right] \quad (4.2.3.1.3)$$

which is readily programmable in open-source software through nested "do-while" loops or similar.

4.2.3.2 Combination 2: Nyquist 241 within Savitsky-Golay 605

Again beginning with the Nyquist algorithm iterating through the smaller (241 point) window, centred at $m = 121$, taken directly from equations 4.2.3.1.1 and 4.2.3.1.3:

$$(t_{ms}, A_{ms}) = \Psi \left[\frac{\sum_{j=m-120}^{m+120} (t_j, A_j) + 7.7621}{2^{17.9129}} \right] \quad (4.2.3.2.1)$$

To be nested within the five-point (polynomial of degree 4) S-G algorithm from equation 3.3.4.2 in Chapter 3:

$$A_{ms} = \frac{1}{35} [-3 \times A_{m-2} + 12 \times A_{m-1} + 17 \times A_m + 12 \times A_{m+1} - 3 \times A_{m+2}] \quad (3.3.4.2)$$

but to make it easier for programming, it simplifies into only three terms as follows:

$$A_{ms} = \frac{1}{35} [17 \times A_m - 3 \times (A_{m-2} + A_{m+2}) + 12 \times (A_{m-1} + A_{m+1})] \quad (4.2.3.2.2)$$

which iterate through the outer (605 point) window centred at $k = 303$ (and so over a 605 point window), the iterative sum – first recalling the nested system above:

$$(t_{ks}, A_{ks}) = {}^{k+i}_{k-i}\Xi\left({}^{m+p}_{m-p}\Theta\right) \tag{4.2.2.4}$$

Where Ξ is S-G iterated, and Θ is Nyquist iterated, equation 4.2.2.4 now becomes:

$$(t_{ks}, A_{ks}) = {}^{k+302}_{k-302}\Xi\left({}^{m+120}_{m-120}\Theta\right) \tag{4.2.3.2.3}$$

But recall that S-G contains a symmetrical 5-point polynomial, which begins at A_{m-2} and ends at A_{m+2}, and hence the iteration of the outer S-G polynomial begins at m, so equation 4.2.3.2.3 reduces to

$$(t_{ks}, A_{ks}) = {}^{k+300}_{k-300}\Xi\left({}^{m+120}_{m-120}\Theta\right) \tag{4.2.3.2.4}$$

and on substitution of Nyquist into 4.2.3.2.4 from equation 4.2.3.1.1:

$$(t_{ks}, A_{ks}) = \Psi\left[{}^{j=k+300}_{j=k-300}\Xi\left[\frac{\sum_{j=m-120}^{m+120}(t_j, A_j) + 7.7621}{2^{17.9129}}\right]\right] \tag{4.2.3.2.5}$$

where, in spite of the apparent complexity of equation 4.2.3.2.5, it can be seen as a simpler series of steps, as:

- Ψ is just a scaling constant, depending on the output of the signal;
- ${}^{k+300}_{k-300}\Xi$ is the iteration of the 5-point Savitsky-Golay polynomial through the 605 point window, taking account of the ends of the polynomial as seen in Figure 3.3.4.1, and re-iterated above.

Again equation 4.2.3.2.5 can be programmed efficiently *in situ* or *post pacto* with a series of nested "do", "do-while" loops or similar.

4.2.3.3 Combination 3: Gauss 381 within Savitsky-Golay 605

With the inner Gauss algorithm iterating through the smaller (381 point) window, centred at $m = 191$ and taken directly from equation 4.2.3.1.2 above:

$$(t_{ms}, A_{ms}) = \sum_{j=m-190}^{m+190} G_j\,(t_j, A_j)$$

In this form, the Gauss algorithm is centred at the k^{th} (central) point; centred at the 191st point, it tends to Gaussian symmetry about that point.

So the iterative composite combining the S-G form of equation 4.2.3.2.4 with the Gauss equation above now looks in it general form as follows:

$$(t_{ks}, A_{ks}) = {}^{k+300}_{k-300}\Xi\left({}^{m+190}_{m-190}\Theta\right) \tag{4.2.3.3.1}$$

where this time, Θ is the iterative form of the Gaussian smoothing algorithm.

Substitution of 4.2.3.1.2 into 4.2.3.3.1 gives:

$$(t_{ks}, A_{ks}) = \Xi_{j=k-300}^{j=k+300}\left[\sum_{j=m-190}^{m+190} G_j\,(t_j, A_j)\right] \tag{4.2.3.3.2}$$

Again equation 4.2.3.3.2, can be seen as a simpler series of steps, where:

- The 191 values of G_j are determined only once initially from

$$G_j(A_k, A_j) = exp\left[-\frac{(A_k - A_j)^2}{2(t_k - t_j)^2}\right] \tag{3.3.5.1}$$

  bearing in mind the condition outlined in §4.2.3.1 above.

- ${}^{k+300}_{k-300}\Xi$ is the iteration of the 5-point Savitsky-Golay polynomial through the 605 point window, taking account of the ends of the polynomial as seen in Figure 3.3.4.1, and re-iterated above;

and again equation 4.2.3.3.2 can be programmed efficiently *in situ* or *post facto* as above.

4.3 Computational and Experimental

Each of the three pairs of nested algorithms outlined above were applied to the two sets of data used in Chapter 3. To provide a logical framework, each nested pair of algorithms was coded in C++ after the fashion shown in APPENDIX 2, and this outline was then rewritten in order to be able to apply it to the data in a Microsoft Excel™ spreadsheet. This *post-facto* application of the programming sequence in a spreadsheet led to smoothed electropherograms of the same appearance as the original individual smoothing algorithms, with no shift in migration times, but with different SNR values.

An initial comparison of SNR using Gauss (381 point window) nested inside Savitsky-Golay (605 point window) enabled firstly a visual comparison of performance, and then numerical comparison as shown below.

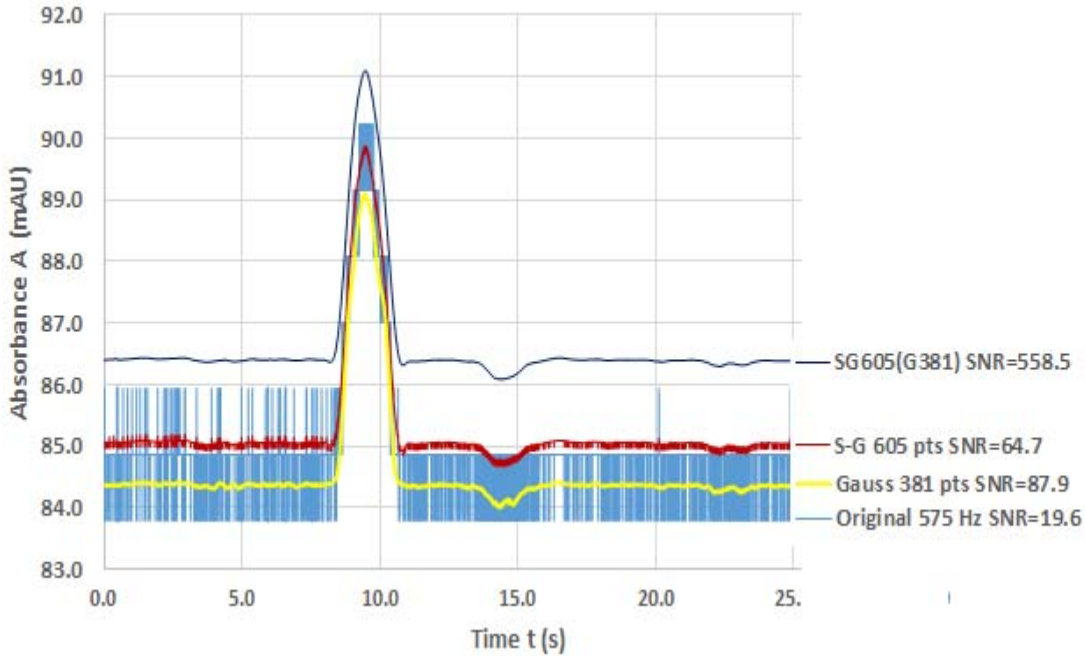


Figure 4.3.1: Overlay of four datasets showing the original 575 Hz DAQ, smoothing using Savitsky-Golay (5-point polynomial) over 605 point window, Gaussian smoothing over a 381 point window and composite smoothing using Gaussian smoothing over a 381 point window nested within Savitsky-Golay (5-point polynomial) over 605 point window.

Further, this qualitative assessment appears also to hold for the multiple peak electropherogram shown in Figure 3.4.2.1 from Chapter 3 §3.4.2.

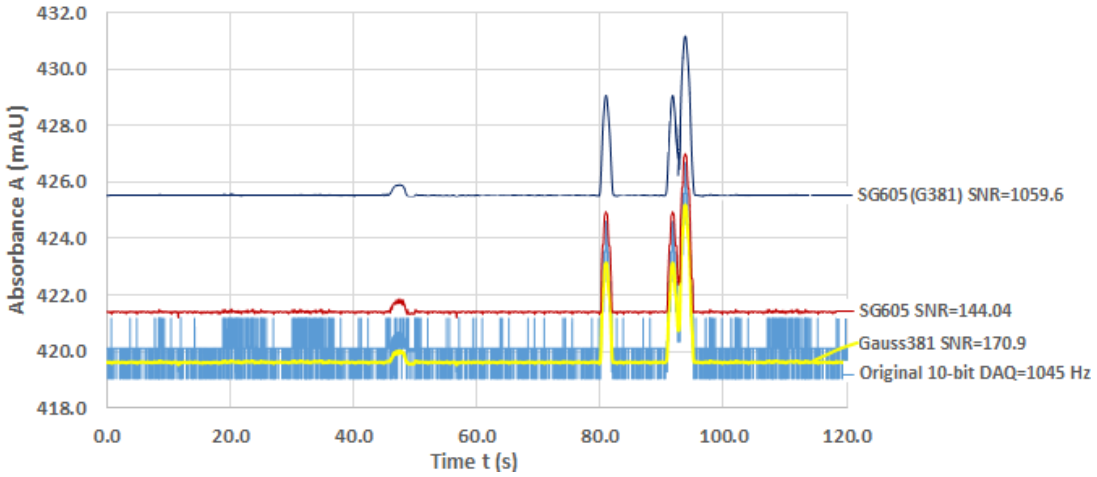


Figure 4.3.2: Overlay of four datasets of the separation of C-102+F+FITC showing the original 1045 Hz DAQ, smoothing using Savitsky-Golay (5-point polynomial) over 605 point window, Gaussian smoothing over a 381 point window and composite smoothing using Gaussian smoothing over a 381 point window nested within Savitsky-Golay (5-point polynomial) over 605 point window.

By way of further illustration, a precise method of determining resolution in the case of compound algorithms is to utilise the vector method derived in Chapter 3. This allowed for a comparison of peak width and resolution from the vector derivative of each of the component algorithms and also from the compound algorithm. The vector derivative of the compound algorithm above in Figure 4.3.2 allows us to get each of the critical points very precisely as follows:

- The segment from 75.0 s to 100.0 s was extracted from the smoothed electropherogram SG605(G381) shown in Figure 4.3.2;
- The vector anchor point of $(t^*, A^*) = (90.0, 422.0)$ was selected, and the vector lengths from (t^*, A^*) to any j^{th} point (t_j, A_j) on the data has each length $|\vec{l}|$ given by

$$|\vec{l}| = \sqrt{(t_j - t^*)^2 + (A_j - A^*)^2} \tag{3.2.3.1}$$

- The rate of change of the vector length $\frac{d|\vec{l}|}{dt}$ reveals the key points of interest, which can can then be calculated automatically in the spreadsheet by embedded formulas. These are shown in Figure 4.3.3 below.

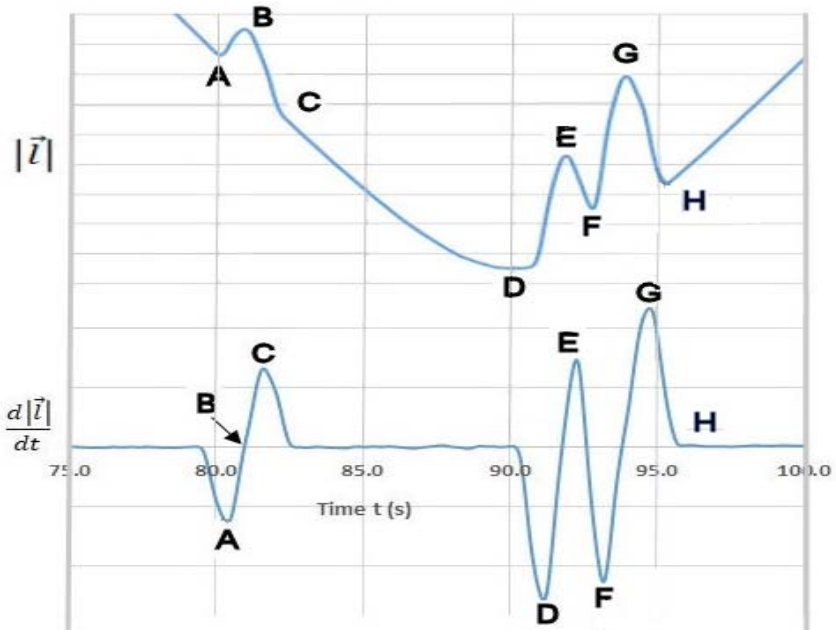


Figure 4.3.3: The plots of $|\vec{l}| \sim$ t and $\frac{d|\vec{l}|}{dt}$ from extracted segment of interest (75.0 s to 100.0 s) from the smoothed electropherogram SG605(G381) of separation of C-102+F+FITC shown in Figure 4.3.2 above.

The datasets, the four electropherograms of Figure 4.3.2 together with their analyses by the vector method illustrated in Figure 4.3.3 and discussed above are set out in APPENDIX 4.

4.3.1 Results

The results, including screenshots of the data and layout of the spreadsheets showing the locations of automated calculation of all of: SNR, Wp, and Resolution can be seen in APPENDIX 4. Also contained in APPENDIX 4 are the electropherograms of each of the component algorithms together with the nested algorithm derived from each pair.

From the smoothing algorithms and their composites chosen in Table 4.2.1.1, the following summary data values were obtained:

Table 4.3.1.1: The results of peak width, SNR and resolution in the cases of single-peak and multiple-peak electropherograms. In the case of composite algorithms, the factor by which the composite SNR is an improvement on the better of the values in the two individual algorithms is shown.

	NYQ (241 pt)	**GAUSS (381 pt)**	**S-G (605 pt)**	**NYQ (241 pt) In S-G (605 pt)**	**NYQ (241 pt) In GAUSS (381 pt)**	**GAUSS (381 pt) In S-G (605 pt)**
Single Peak SNR (factor of larger SNR)	89.60 Max 92.66	87.18 Max 91.22	93.31 Max 91.12	559.86 (6.18)	448.00 (4.55)	373.24 (3.46)
Single Peak Wp (factor of larger Wp)	2.585 Max 3.896 @1023 pts	2.476 Max 2.968 @1023 pts	2.601 Max 2.924 @1023 pts	2.592 (0.996)	2.491 (0.964)	2.488 (0.956)
F-FITC SNR pk1 (factor of larger SNR)	413.28	361.8	428.35	2570.10 (6.15)	2066.40 (4.38)	1713.40 (3.42)
F-FITC Wp pk1 (factor of larger Wp)	2.537 Max 3.935 @1023 pts	2.501 Max 2.974 @1023 pts	2.690 Max 2.917 @1023 pts	2.677 (0.995)	2.433 (0.959)	2.561 (0.952)
F-FITC Resolution peaks 2&3	0.89 Res=0.66 @1023 pts	0.75 Res=0.59 @1023 pts	0.84 Res=0.76 @1023 pts	0.92	0.91	0.87

4.3.2 Analysis of Results

In an analysis of the application of all three pairs of nested algorithms to both electropherograms, peak width in each case was determined by using the vector length method outlined in §3.2.3 of Chapter 3.

It was noted in Chapter 3 that there was a change in order of performance between the best 3 algorithms depending on whether a single-peak or multiple-peak electropherograms were considered. (Refer to Table 3.4.1.1, Table 3.4.2.1 and APPENDIX 4: Table Apx 4.1)

The performance summary in APPENDIX 4, Table Appx 4.1 shows a surprising GMS performance on Resolution (score: 4.795) - in fact above that of the Gaussian smoothing (score: 2.797). The precise reason for this is not clear at this time, but along with similar matters will need to be examined in some depth. Nevertheless, it is

interesting to note that both Gaussian and GMS are statistical treatments of the problem of smoothing.

In the case of the three pairs of nested algorithms studied above however, such an anomaly did not appear – there was a clear order in each case on every electropherogram on the criteria of peak width and SNR in the case of the single peak electropherogram, and peak width, SNR and resolution in the multiple-peak electropherogram. It became apparent that the best performing combination in these electropherograms was S-G_{605}(Nyq_{241}), which seems to deliver an improvement in SNR by a factor of about 6.2 times.

4.4 Conclusions

The technique of this Chapter was to use the compound function $f \circ g$ or $f(g(t))$ where two algorithms of different underlying mathematical structure are applied to the same dataset at the same time. This means that noise reduction occurs simultaneously by two different methods, and signal clarification is dealt with by two different underlying functional models, a mathematical process known as *mollification*.[15]

In this chapter, criteria were established whereby composite algorithms could be derived using the mathematical frameworks of Chapter 3. These ideas were extended to enable valid composite pairs of algorithms to be coded in C++ and Microsoft Excel™ and applied to signal smoothing.

For very low DAQ (DAQ < 100 Hz) the methodology can still be applied. At such low DAQ, $\sigma_{At} \rightarrow \sigma_A$ because errors in $t \rightarrow 0$ as jitter becomes insignificant at these low rates. As stated in § 3.1.4 above, slower DAQ does lend itself to phase and signal shift when using window-based smoothing. All other mathematical treatments remain unaffected.

The three pairs of nested algorithms developed were in this case able to be directly ranked from the data outcomes using the criteria used in Chapter 3, viz. signal-to-noise ratio, peak width and resolution. It turned out that the vector method for resolving peak width could be extended to a precise determination of several peak positions and also used to precisely resolve closely aligned peaks. It seems as if this method has never been used in chemistry to resolve electropherograms, although there is a great deal of innovative related work which has recently been done by Kalambet.[18]

The question now arises as to whether this method compromises peak shape to any extent, and so possibly compromises a reliable way of integrating to find peak area. In the work of Dubský, Dvořak et.al,[19] any compromise in peak shape would also compromise the determination of migration time from the Haarhoff–van der Linde (HVL) function and its application to peak geometry.

This technique enables the problem of a slowed process (due to nesting algorithms) of *in situ* processing to be resolved.

4.5 References:

1. Hu, B., Gosine, R.G., A New Eigenstructure Method for Sinusoidal Signal Retrieval in White Noise Estimation and Pattern Recognition. *IEEE Transactions on Signal Processing* **1997,** *45* (12), 3073-3083.

2. Yi, F. S., Taek, L.S., Tae, Y.U., Integrated Particle Rauch-Tung-Striebel Backward Smoothing for Single Target Tracking in Clutter. In *15th International Conference on Control, Automation and Systems*, BEXCO: Busan, Korea 2015; pp 393-398.

3. Schnoll-Bitai, I., Vega, J., Mader, C., Estimation of the band broadening parameters in single detection size-exclusion chromatography: a comparative study of various column combinations. *Anal Chim Acta* **2007,** *604* (1), 9-17.

4. Bernabé-Zafón, V., Torres-Lapasió, J. R., Ortega-Gadea, S., Simó-Alfonso, E. F., Ramis-Ramos, G., Capillary electrophoresis enhanced by automatic two-way background correction using cubic smoothing splines and multivariate data analysis applied to the characterisation of mixtures of surfactants. *Journal of Chromatography A* **2005,** *1065* (2), 301-313.

5. Arondo, J.-L. R., Muga, A., Castresana, J., Goñt, F.M., Quantitative Studies of the structure of proteins in solution by Fourier Transform Infra-Red Spectroscopy. *Progress in Biophysics and Molecular Biology* **1993**, 23-56.

6. Liu, B. F., Sera, Y., Matsubara, N., Otsuka, K., Terabe, S., Signal denoising and baseline correction by discrete wavelet transform for microchip capillary electrophoresis. *Electrophoresis* **2003,** *24* (18), 3260-3265.

7. Szymanska, E., Markuszewski, M. J., Capron, X., van Nederkassel, A. M., Heyden, Y. V., Markuszewski, M., Krajka, K., Kaliszan, R., Increasing conclusiveness of metabonomic studies by chem-informatic preprocessing of capillary electrophoretic data on urinary nucleoside profiles. *J Pharm Biomed Anal* **2007,** *43* (2), 413-420.

8. Enke, C. G., Nieman, T.A. , Signal-to-Noise Ratio Enhancement by Least-Squares Polynomial Smoothing. *Analytical Chemistry* **1976,** *48* (8), 705-712.

9. Doucet, A., Andrieu, C., Iterative Algorithms for State Estimation of Jump Markov Linear Systems. *IEEE Transactions on Signal Processing* **2001,** *49* (6), 1216-1227.

10. Sugimoto, M., Hirayama, A., Ishikawa, T., Robert, M., Baran, R., Uehara, K., Kawai, K., Soga,T.,Tomita, M., Differential metabolomics software for capillary electrophoresis - mass spectrometry data analysis *Metabolomics* **2010,** *2010* (6), 27-41.

11. Briers, M., Doucet, A., Maskell, S., Smoothing algorithms for state–space models. *Annals of the Institute of Statistical Mathematics* **2009,** *62* (1), 61-89.

12. Kitagawa, G., Monte Carlo Filter and Smoother for Non-Gaussian Non-linear State Space Models. *Journal of Computational and Graphical Statistics* **1996,** *5* (1), 1-25.

13. Fearnhead, P., Wyncoll, D., Tawn, J., A sequential smoothing algorithm with linear computational cost. *Biometrika* **2010,** *97* (2), 447-464.

14. Solis, A. R., M.; Campiglia, A. D.; Sojo, P., Accelerated multiple-pass moving average: A novel algorithm for baseline estimation in CE and its application to baseline correction on real-time bases. *Electrophoresis* **2007,** *28* (8), 1181-1188.

15. Beatson, R. K., Bui, H.-Q., Mollification Formulas and Implicit Smoothing. *Research Report UCDMS; Christchurch, NEW ZEALAND* **2003,** *19* (2), 1-11.

16. Madden, H. F., Comments on the Savitzky-Golay Convolution Method for Least-Squares Fit Smoothing and Differentiation of Digital Data. *Anal. Chem* **1978,** *50* (9), 1383-1386.

17. Bangliang, S., Yiheng, Z., Lihui, P., Danya, Y., Baofen, Z., The use of simultaneous iterative reconstruction technique for electrical capacitance tomography. *Chemical Engineering Journal* **2000,** *77*, 37-41.

18. Kalambet, Y., Kozmin, Y., Mikhailova, K., Nagaev, I., Tikhonov, P., Reconstruction of chromatographic peaks using the exponentially modified Gaussian function. *Journal of Chemometrics* **2011,** *25* (7), 352-356.

19. Dubsky, P., Dvorak, M., Mullerova, L., Gas, B., Determination of the correct migration time and other parameters of the Haarhoff-van der Linde function from the peak geometry characteristics. *Electrophoresis* **2015,** *36* (5), 655-661.

5 Application of Nested Compound Smoothing Algorithms to fast Electrophoretic Data Acquisition

5.1 Introduction: Signals with High DAQ and High Noise

The principal aim of this chapter is to extend the application of techniques developed in Chapters 3 and 4 to raw datasets with much higher DAQ and noise than those previously analysed. The algorithms used here were previously validated in Chapter 3 using simulated data, and then in Chapter 4 using electropherograms of low-resolution/high DAQ in real-time separations with varying S/N, noise and resolution.[1] In the previous chapters, algorithms operate serially with each new electropherogram having to be analysed separately. Within each spreadsheet, every new smoothing window is delineated by a separate tab. As the signals begin to appear from under the noise, windows are established which encompass each peak and formulas to calculate standard deviation, peak height, and peak width can then be inserted[2] into each tab allowing these parameters to be calculated as soon as the data is pasted.

5.1.1 Some Notes on the Nature of High DAQ Signals

An initial assessment of high DAQ signals with high noise and low SNR often leads an experimenter to take a cautious approach, leading back to either tweaking or modifying the detector attributes and sensitivity settings,[3] the architecture of the instrument itself,[4] or to modify the chemistry in order to coax a better signal or improve detection limits from which some useful conclusions may lead.[5] Some realities of high DAQ/high noise datasets seem to be the following:

- Errors may occur during DAQ – in any ADC converter that uses smoothing (sometimes referred to as "noise shaping"[6]) to improve resolution, the frequency response of the converter is only flat to its maximum useful bandwidth, beyond which the converter may resonate, (an eigenvector effect) and cause a signal to be amplified before ADC. This adds complexity to the signal acquisition; such signals then must be attenuated in the digital filter[7] to prevent aliasing. This is particularly apparent in the *in situ* application of the Nyquist algorithm where, if the applied signal contains major signal components such as harmonics or high noise, the converter may overload on some of the signal data.

- Measurements are often carried out in a complex sample background, concurrently presenting multiple precursors which may generate interfering traces.[3]
- More information is contained in a high DAQ signal than in a low DAQ signal;
- Information which may be visually hidden from a high-noise/low signal trace actually contains information such as the position of a baseline as outlined in Chapter 3, §3.2.1;
- Data density itself within a particular interval is a source of information.

High signal density techniques are routinely used in the application of optical microscopes to samples of molecular size within living cells, where increased DAQ minimizes errors in the detection signal resulting from cellular or object movement. In such techniques, high data density obtained from different simultaneous laser absorbance signals is used to derive accurate information about intra-cellular molecular locations and interactions.[8]

In high-density DAQ signal analysis, algorithms used are often bespoke - derived specifically for a particular circumstance, to selectively subtract irrelevant optical, electrostatic or similar extraneous noise.[9] Flexibility can be achieved by using:

- plug-in technology;
- deferred execution (*post-facto* and usually sequential) although such a method is slow (requiring time for file analysis) and then data analysis becomes the rate-limiting step in an experimental procedure. Currently, such a method also fails to produce comprehensive reports across a dataset; single reports must be combined to show correlations[1]; and/or
- implicit invocation of different algorithms during data acquisition (*in situ*). Such processes may be derived in different ways with different levels of accuracy and reliability, and are necessarily subject to validation and testing before use.[10]

5.1.2 Two Signals studied and Processed

To increase the DAQs which were used earlier, two raw datasets for high-DAQ and high noise electropherograms were acquired from independent sources. They are as follows:

1. The first was acquired using an EMANT300™ portable data acquisition system; it was used to acquire a 3 kHz signal at resolution of 22 bits ADC.
2. The second was acquired using the same EMANT300™ portable data acquisition system; it was used to acquire a 6 kHz signal at resolution of 16 bits ADC.

In both cases, data was not only high noise but very low SNR, so that any underlying signals were completely obscured.

5.2 Devising a Methodology for Signal Processing

Based on the methods tested and used in Chapters 2, 3 and 4, it was decided to take Signal 1 as a test case to try to develop a methodology in order to do the following:

- attempt to reveal any underlying signals as far as possible, including signals of lowest SNR within the dataset;
- pick a clear signal of interest and use it to apply the optimal smoothing methods described in chapter 4 in order to achieve maximum SNR;
- compare the optimal parameters for maximised SNR from Chapter 4 with those achieved in Signal 1;
- use the same signal and apply optimal smoothing methods to find the relationship between the smoothing parameters and peak width *Wp* around maximum SNR;
- compare the optimal parameters for maximised *Wp* from Chapter 4 with those achieved in Signal 1;
- use two closely adjacent peaks (if and where available) in order to determine resolution around the parameters which give maximum SNR.

5.2.1 Signal 1: DAQ = 3 kHz; 22-bit ADC with Method Development

Apart from the data acquisition frequency and chip resolution, the only information given about this signal is that the injection time was 5.0 s. The first step was then to obtain a visual representation of the signal as follows:

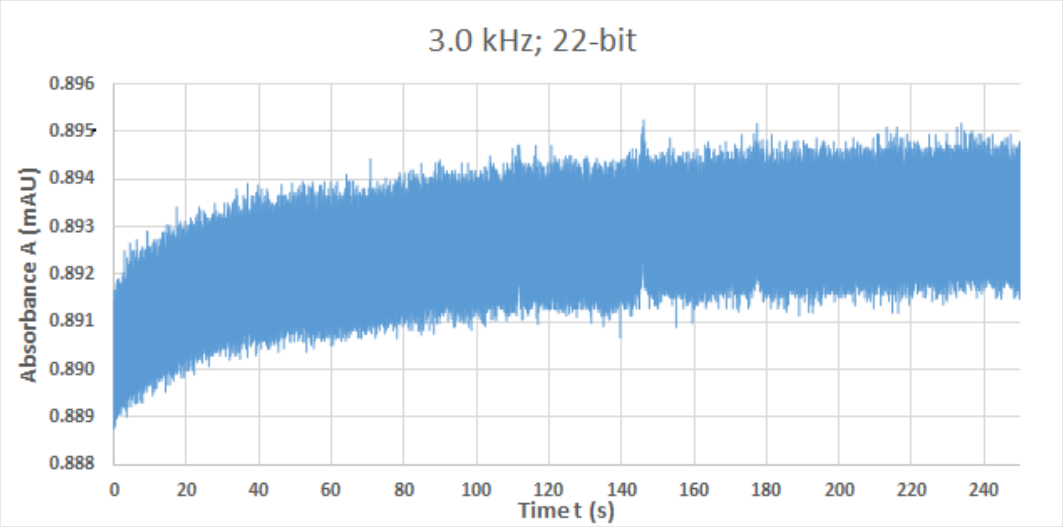


Figure 5.2.1.1: Original electropherogram of the first dataset with DAQ = 3 kHz and 22-bit resolution, showing times and absorbance obtained from commercial (EMANT 300™) system and instrument controlled by LabVIEW™ software. The only information known was a 5.0 s hydrodynamic injection of low concentration analyte.

In an electropherogram with such high data density, it is first necessary to see whether any underlying signals are detectable; a common method for detecting such signals is to use a derivative test, often also a good for identifying closely adjacent peaks at low resolution.[11] Such a derivative test is quite powerful if used correctly, but in a case where SNR $\rightarrow$0, the first derivative test is of no use when directly applied.

In Figure 5.2.1.2 below, the first derivative as a means for identifying any peak seems of no value; unsurprising in view of the earlier illustration of such an attempt shown in Figure 3.2.2 which appears in §3.2.2 of Chapter 3.

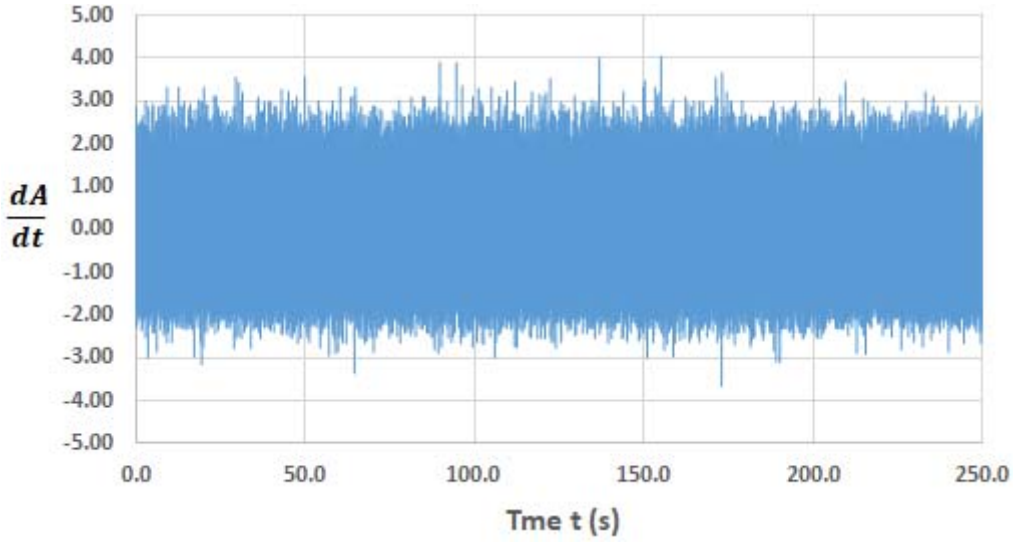


Figure 5.2.1.2: Direct first derivative of the raw signal demonstrating the inability of a high noise signal to be revealed by such a direct method.

Before applying any of the nested algorithms, smoothing was first attempted by simply using the Gaussian algorithm alone as adapted and derived from Chapter 3 with a 153 point smoothing window. This technique revealed underlying signals as shown below. In fact, any of the 5 common smoothing methodologies from Chapter 3 can be used for a variety of low SNR/high frequency signals – for example, in the work of Gallo, Capozzi *et al*, Savitsky-Golay is used in this way for a first visualisation of signals of interest.[11]

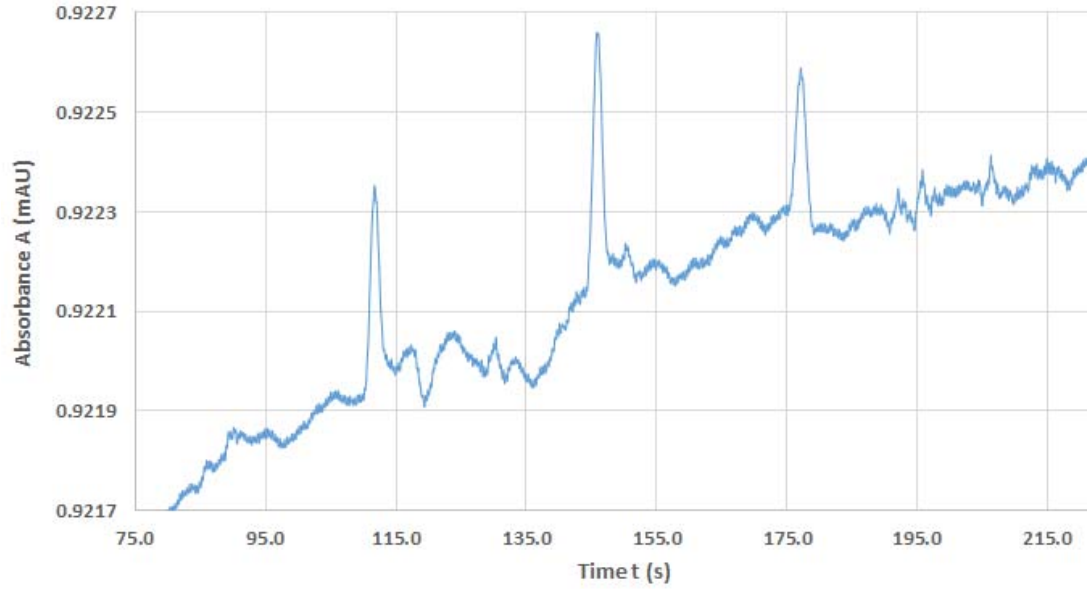


Figure 5.2.1.3: Electropherogram of the first dataset with DAQ = 3 kHz and 22-bit resolution, and initial smoothing by *post-facto* iterative application of the Gaussian algorithm only, using a 281 point window which is 100 points smaller than the optimum of 381 points determined in Chapter 3.

For this 3 kHz signal, the central peak from around 144.0 s to 150.0 s was chosen for subsequent analysis. Since there is good separation between the three peaks on this electropherogram, analysis did not include any reference to peak resolution.

Now it makes much more sense to apply the first derivative to this initial roughly-smoothed electropherogram, and allows for the more precise determination of the position of the endpoints of the central peak identified above.

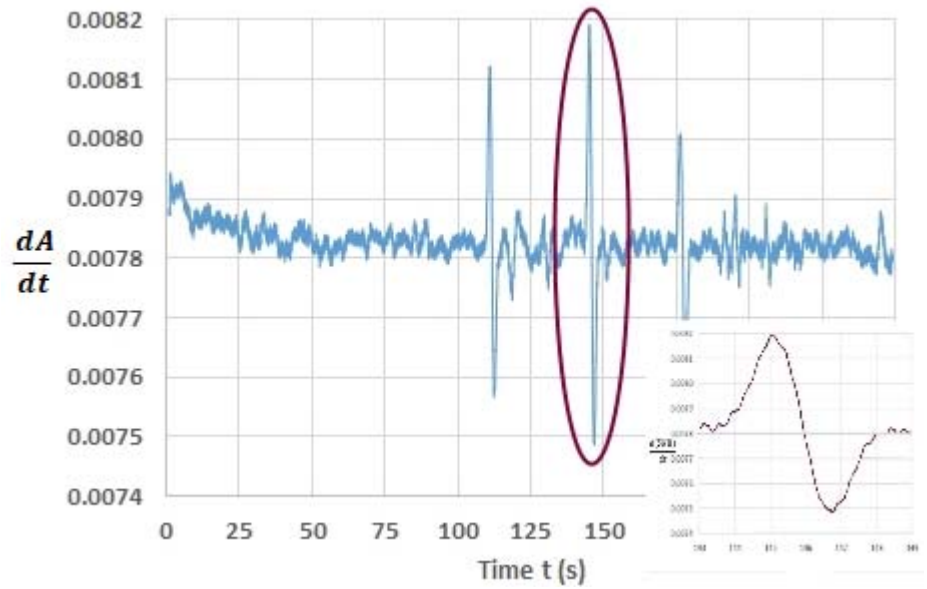


Figure 5.2.1.4: First derivative plot of electropherogram of the first dataset with DAQ=3 kHz and 22-bit resolution, after initial smoothing using a 281 point smoothing window shown above. This allows more precise visual determination of peak position and endpoints in the signal of interest.

Initially all three of the top algorithms identified in Chapter 3 were applied to this first dataset over window sizes from 89 points to 1023 points. Reassuringly the same optimum values of SNR were obtained at around the same optimal window sizes (RSD< 0.5%) identified in Chapter 3 in all three cases, which seems to suggest that these values may possibly also hold for higher DAQ rates.

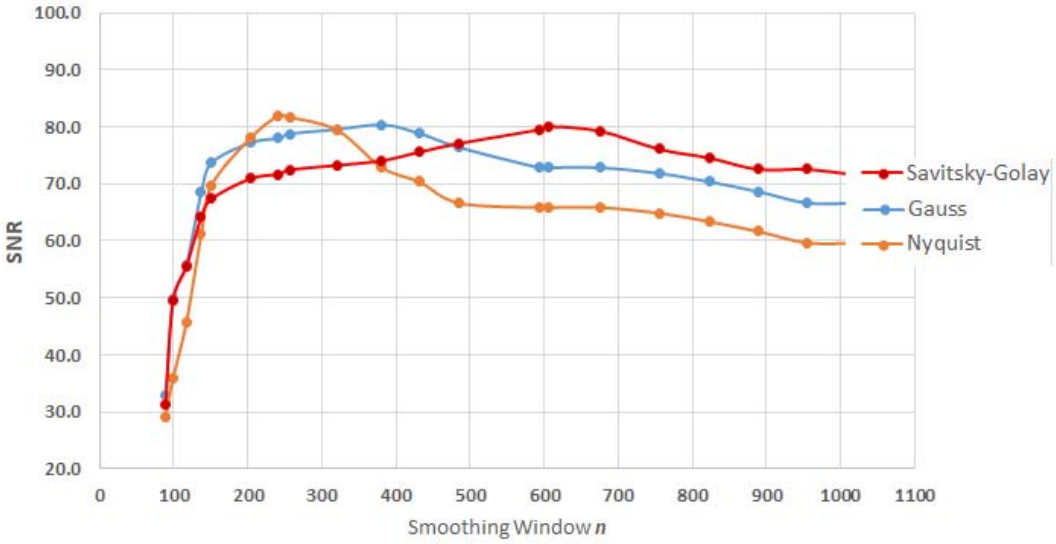


Figure 5.2.1.5: The relationship between smoothing window size and SNR for the three best-performing algorithms from Chapter 3. In all three cases, the maxima occur around the same values as before, *viz* Savitsky-Golay 609 points; Gauss 383 points; Nyquist 239 points.

All three of the nested algorithms identified in Chapter 4 were applied to this electropherogram, but in order to identify the maximum SNR, the following procedures were followed:

1. the larger window was kept at its maximised constant value (S-G 609, Gauss 383, Nyquist 239);
2. the smaller internal window was iterated through values at 10 point intervals for window sizes 100 points on either side of its optimal smoothing value (Gauss from 283 to 483, and Nyquist from 139 to 339);
3. the SNR values obtained were then plotted against the internal window size of the smaller internal window.

Table 5.2.1.1: Combinations of maximised external and internal (iterative) smoothing windows used in the SNR and Peak Width analysis of the dataset of Fig. 5.1.2.1.1 shown above.

		OUTER ALGORITHM WINDOW		
		NYQ (239 pt)	**GAUSS (383 pt)**	**S-G (609 pt)**
INNER ALGORITHM WINDOW	**NYQ (139 pt - 339 pt)**	N	G383(N139-339)	SG609(N139-339)
	GAUSS (283 pt - 483 pt)	N	N	SG609(G283-483)
	S-G (509 pt - 709 pt)	N	N	N

It turns out that incidentally a future possible use of this technique may well be in the reconstruction of a signal from very sparse or interrupted datasets.[12]

To be fair and to yield more comprehensive results, it would be better to take both internal and external smoothing windows over a number of intervals on either side of their optimum values and look at results for all possible valid combinations, bearing in mind the restrictions outlined in Rules 1 to 4 stated in §4.2.1 from Chapter 4. Such a more detailed approach has been left for future work and is outlined in Chapter 6.

The first pair of algorithms used on this dataset was Nyquist(139 – 339) with windows iterated successively at 10-point intervals inside Gauss(383). This established the workability of this approach, and the other two combinations then followed according to the protocols outlined in Table 5.2.1.1.

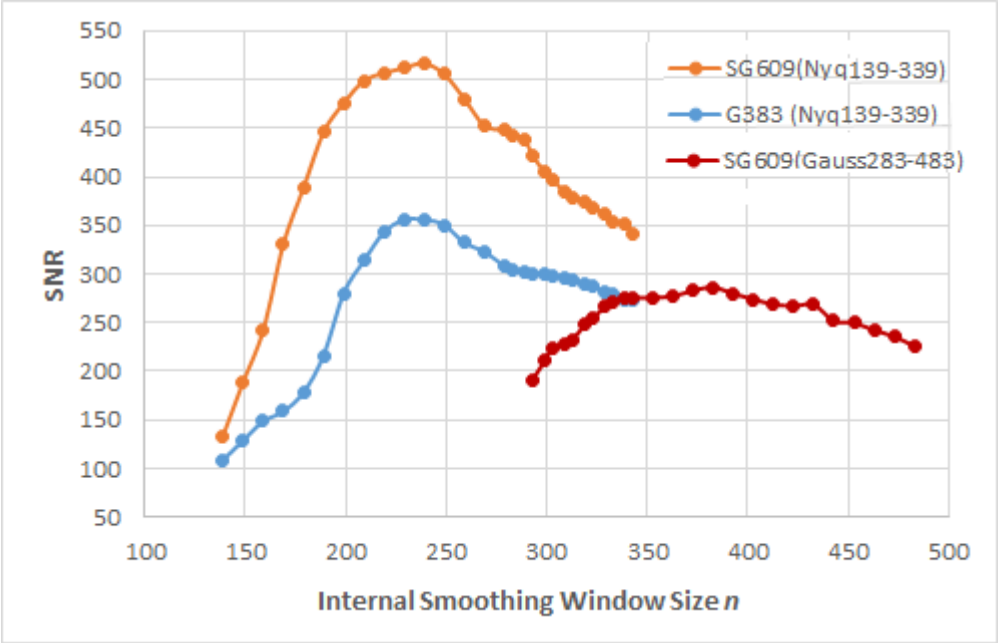


Figure 5.2.1.6: Illustration of the relationship between SNR and three pairs of nested smoothing algorithms. The smaller internal algorithm window is iterated successively at 10-point width increments as shown in Table 5.1.2.1.1inside the fixed outer optimal window sizes.

Now, for the method used to determine peak width in every case, the same three conditions were applied to determine peak width by using the vector method described in Chapter 3. The full method outline is described here again for explanatory completeness as it is applied.

All three of the nested algorithms identified in Chapter 4 and used above on Signal Dataset 1 were again applied. To identify the variations in peak width on either side of the maximum SNR, the following procedures were again followed:

1. the larger window was kept at its maximised constant value (S-G 609, Gauss 383) as shown in table 5.1.2.1.1;
2. the smaller internal window was iterated through values at 10 point intervals for window sizes 100 points on either side of its optimal smoothing value (Gauss from 283 to 483, and Nyquist from 139 to 339);
3. the vector method was then applied in each case at 10-point increments to determine peak width as illustrated on the peak extracted from Figure 5.1.2.1.3 below:

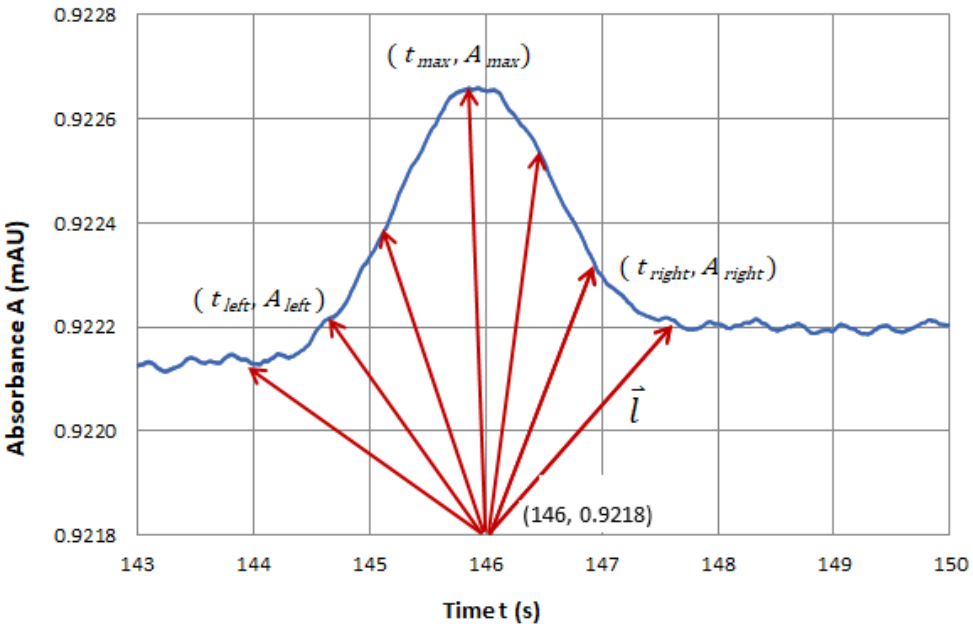


Figure 5.2.1.7: Application of the vector method from Chapter 3 for determination of a peak width at baseline. The anchor point (t^*, A^*) is the point (146 , 0.9218)

As a first step, the vector $\vec{l}$ from (t^*, A^*) to any j^{th} point (t_j, A_j) on the data has length $|\vec{l}|$ given by:

$$|\vec{l}| = \sqrt{(t_j - t^*)^2 + (A_j - A^*)^2} \tag{3.2.3.1}$$

and was applied by extracting the region of interest from the original dataset and applying the values of length $|\vec{l}|$ directly:

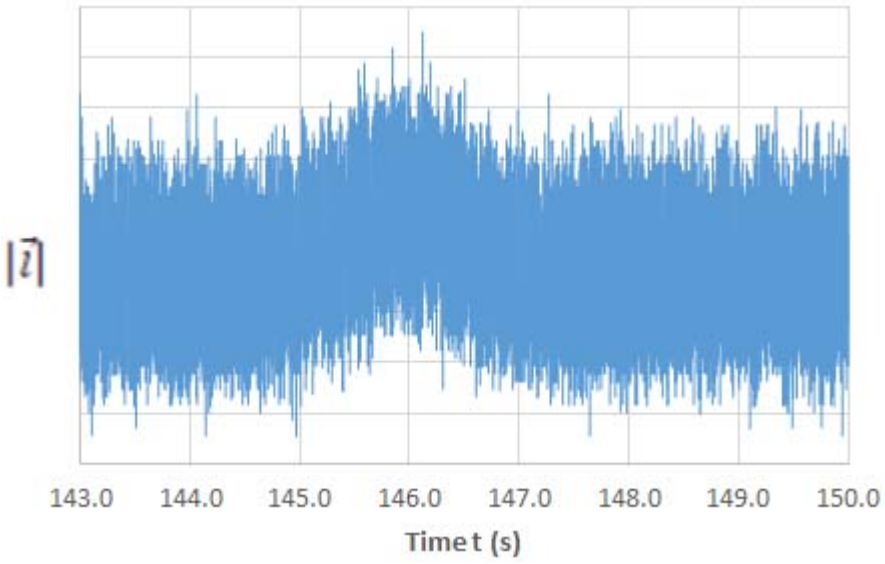


Figure 5.2.1.8: The relationship between vector length $|\vec{l}|$ and time for the extracted section of raw data of interest.

This illustrates the statement from Chapter 3 that a plot of $|\vec{\boldsymbol{l}}|$~t is of no use in such a noisy signal, because noise in the regions around (t_{left} , A_{left}) and (t_{right} , A_{right}) remain noisy and so the point values themselves are difficult to determine visually without more significant techniques. Now using the technique developed in Chapter 3, and §3.2.3 from the derivative of equation (3.2.3.1) in Chapter 3:

$$\frac{d|\vec{\boldsymbol{l}}|}{dt} = \frac{(\boldsymbol{t_j} - \boldsymbol{t^*}) + \frac{dA}{dt}(\boldsymbol{A_j} - \boldsymbol{A^*})}{|\vec{\boldsymbol{l}}|} \tag{3.2.3.2}$$

Similarly, it can be seen from the figure below that the 1st derivative of $|\vec{\boldsymbol{l}}|$~t is of little use in determining peak width because of the very high noise. However it is apparent that such a technique could work because it is clear that $\frac{d|\vec{l}|}{dt}$ shows a fluctuation.

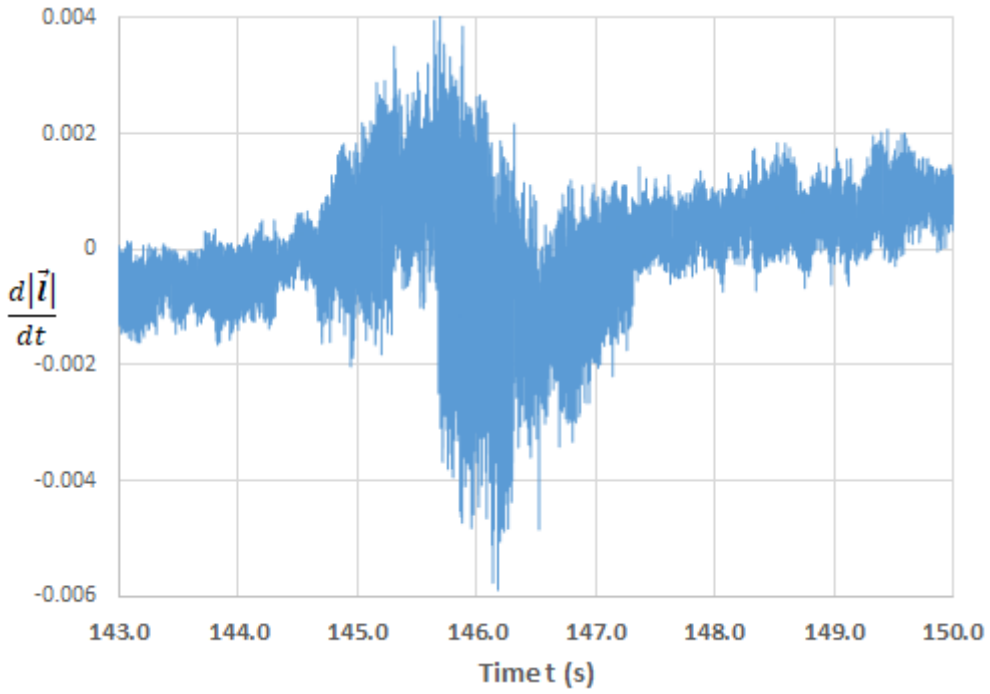


Figure 5.2.1.9: The relationship between the derivative of vector length $\frac{d|\vec{l}|}{dt}$ and time for the extracted section of $|\vec{\boldsymbol{l}}|$~t in Figure 5.2.1.8.

A comparison with Figure 3.2.3.2 in Chapter 3 makes the point quite clearly. The question now arises as to how this matter of peak width may be resolved. It was decided that smoothing with the Nyquist Theorem using a 103 point window would be the best way to resolve this issue because of the snapshot resolution structure implicit in the Nyquist Theorem, which would not require a correcting shift on the time axis. The outcome of this decision is shown in Figure 5.2.1.8 below.

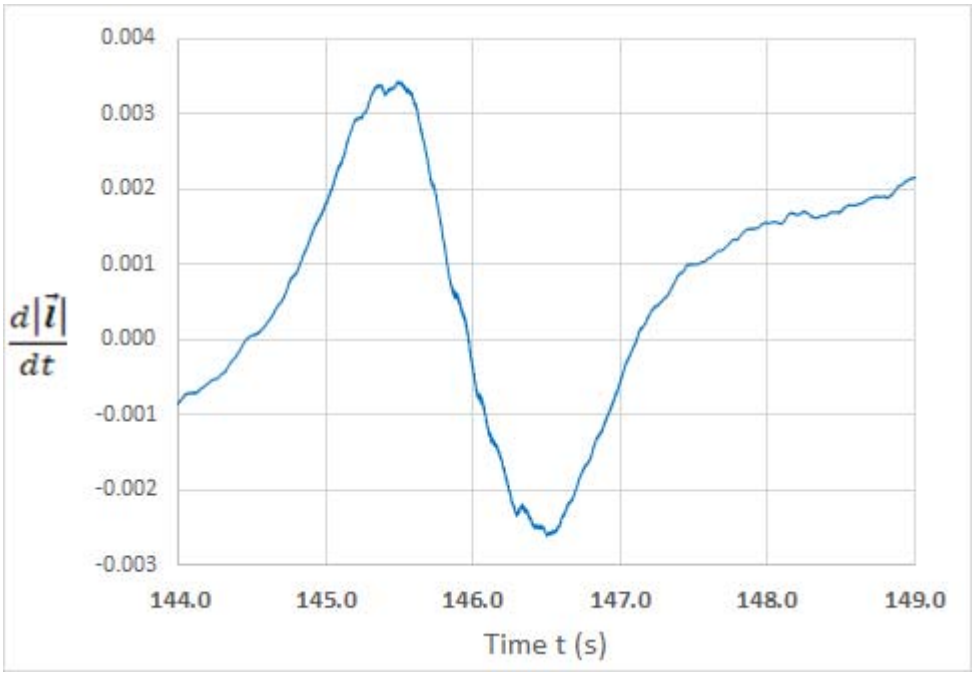


Figure 5.2.1.8: The relationship between derivative of vector length $|\vec{l}|$ and time for the extracted section of $|\vec{l}|$~t after smoothing the data in Figure 5.1.2.7 with a 103 point Nyquist smoothing window.

The application of data-based functions such as VLOOKUP, INDEX, MAX and MIN in Microsoft Excel™, allow for unambiguous determination of peak position, baseline, peak height and peak width by identifying precise positions of (t_{left} , A_{left}) , (t_{max} , A_{max}) and (t_{right} , A_{right}) from the function in Figure 5.1.2.1.7 and Figure 5.1.2.1.8 above. This in turn allows for a detailed comparative analysis of performance of the composite algorithms by using the methods outlined in Chapter 3 and explained further in Chapter 4;

4. the peak width values obtained by this vector method were then plotted against the window size of the smaller internal window.

Data values and graphical visualisations, together with the Excel™ commands can be seen in APPENDIX 5.

5.2.2 Extraction of Stepwise Method

It is important to note initially that in standard data acquisition cards, it is not possible to increase both ADC and DAQ for the same dataset; an increase in rate of data acquisition must always be accompanied by a decrease in resolution. This relationship was introduced in Chapter 2, §2.2.1, revisited again in Chapter 3 in Figure 3.3.6.1 and Figure 3.3.6.2.

Paragraph 5.2.1 above has established the workability of nested algorithms as a smoothing method, and also introduced a possible sequence for revealing and analysing obscured signals in a high noise environment. Consequently, these spreadsheet techniques are now clarified into a simple system from which key values and peak characteristics can be extracted stepwise as follows:

Step 1: Examine the visual dataset; if there are easily identifiable visible signals of interest, apply the derivative test to reveal their locations. If noise is too large to give any meaning to SNR, then apply Step 2.

Step 2: Apply any one of the smoothing algorithms: Nyquist, Savitsky-Golay, or Gauss to the dataset at a window size which reveals the smallest signal of interest at SNR $\cong$ 3.0 by visual inspection. This will form the baseline data for analysis.

Step 3: Extract only the segment of interest for analysis from the dataset. In high DAQ/high noise data, analysis is very time expensive, and extraction of such a segment can significantly reduce computing time. At this stage, the raw data can be adjusted by a scaling factor so that it is of the same order of magnitude as the time scale. This can be done without loss of resolution or analytical validity, and allows a meaningful computation of the vector length to determine peak width.

Step 4:

Apply the data-based functions such as VLOOKUP, INDEX, MAX and MIN in Microsoft Excel™ to the extracted dataset for unambiguous determination of peak position, baseline, peak height and peak width by identifying precise positions of (t_{left} , A_{left}) , (t_{max} , A_{max}) and (t_{right} , A_{right}) from the functions . This in turn allows for a detailed comparative analysis of performance of the composite algorithms by using the methods outlined in Chapter 3 and explained further in Chapter 4.

5.3 Detailed Analysis of High-Noise/Low SNR Electrophoresis Dataset with DAQ = 6 kHz and 16-bit resolution

Step 1: Examination of the second low resolution/high noise dataset is illustrated below:

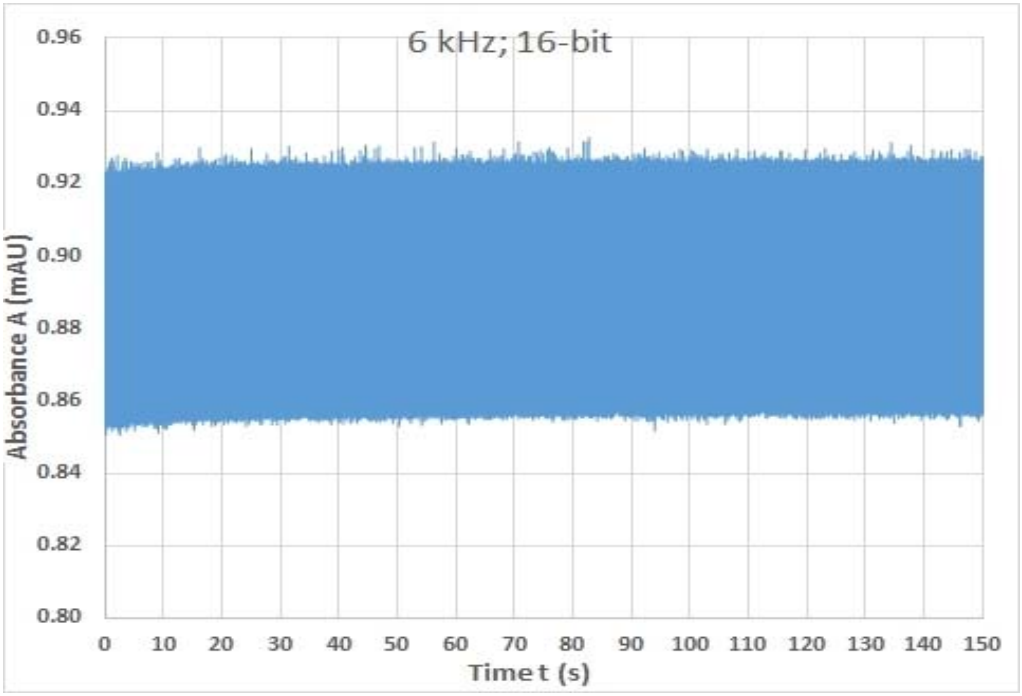


Figure 5.3.1: Original electropherogram of the second dataset with DAQ = 6 kHz and 16-bit resolution, showing times and absorbance obtained from commercial (EMANT 300™) system and instrument controlled by LabVIEW™ software. The only information again known was 5.0 s hydrodynamic injection of low concentration analyte.

The first derivative test to reveal the presence of underlying signals applied in §5.2.1 above is omitted because of the exaggerated noise principle shown in Figure 5.2.1.2.

Step 2: Apply Nyquist (131 point window) to the dataset to reveal signals.

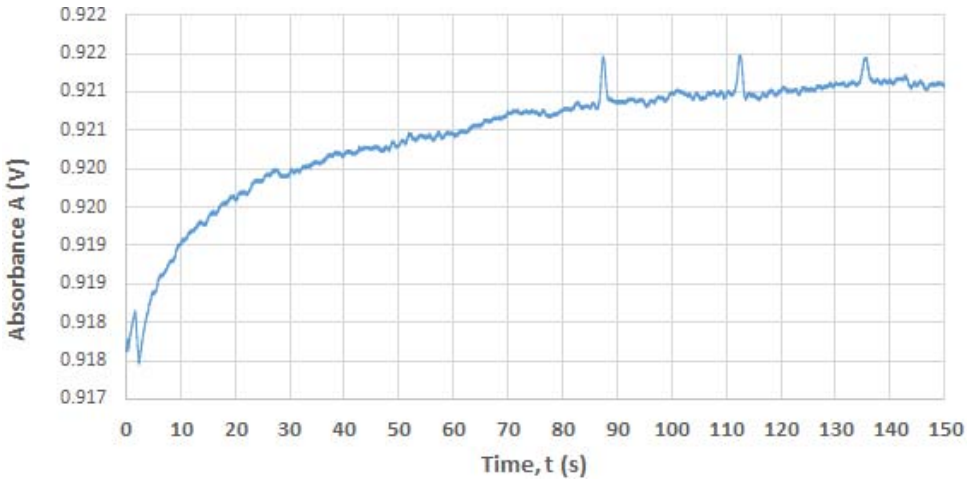


Figure 5.3.2: Electropherogram of the second dataset with DAQ = 6 kHz and 16-bit resolution, and initial smoothing by *post-facto* application of the Nyquist algorithm only, using a 131 point window which is 110 points less than the approximate optimum window size suggested in Chapter 3.

The complete electropherogram in Figure 5.3.2 above shows visually that Nyquist also seems to be a reasonable way to initially reveal peaks of interest. The electropherogram contains around 10^6 data points, but to reveal the smallest signal of interest, it is necessary to proceed to Step 3.

Step 3: From this initial visualisation it is possible to extract the subset of interest from 80 s to 150 s, and then

- use a Nyquist smoothing window as small as possible to give the smallest signal of interest at SNR ≅ 3.0 by visual inspection.
- without loss of data, re-scale the Absorbance (A) axis by an appropriate factor (in this case, ×100) to put Time (t) axis and Absorbance (A) axis values in the same order of magnitude, which allows for a clearer image of the vector processing without the function having very sharp, off-scale peaks or very small, low-amplitude peaks, either of which make visual assessments difficult.

This data extract then forms the baseline data for analysis, as shown below.

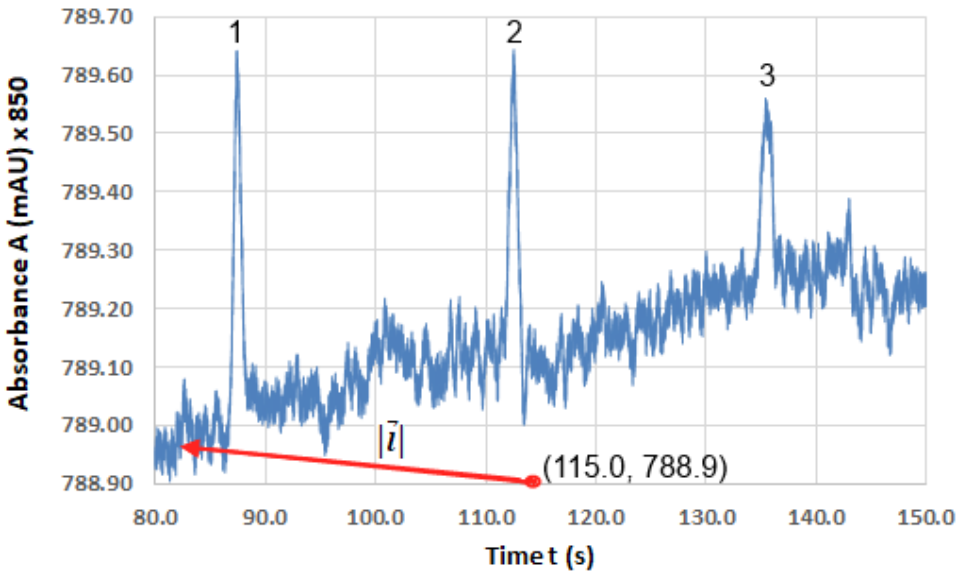


Figure 5.3.3: Re-scaled data extract from the electropherogram of Figure 5.3.2 above. This smoothed segment uses the Nyquist method with 131 point smoothing window. . The anchor point (t^*, A^*) is the point (115.0 , 788.9). The smallest peak number 3 has in SNR = 3.02 and peak width Wp = 1.65 s. The spreadsheet showing the determination of SNR and Wp for all three peaks can be seen in APPENDIX 5.

Since this initial smoothing is done by the Nyquist method, there is no significant shift in migration times as smoothing is achieved by increased resolution. Adjustment of raw absorbance data by a scaling factor is done, also without loss of resolution or analytical validity, because the adjustment is linear and proportional by a factor of 850 in the above case, and does not affect SNR as peak height and noise remain in the same proportion; there is also no horizontal shift and so migration times in peak widths are not affected.

From the dataset extract above, a detailed analysis using each of the set of nested algorithms is applied to the dataset, where a pre-programmed spreadsheet is used to determine performance on all relevant criteria. The method outlined in §5.2.1 will be pursued in detail including the internal nesting technique where the internal algorithm window will be cycled through sizes at 10-point intervals on either side of the optimum value in order to determine the point of maximal performance. This is an extension of the earlier development of pre-programmed spreadsheets first proposed and outlined in Chapter 2, §2.1.4. The datasets and layout of these spreadsheets can be seen in APPENDIX 5.

Step 4: At this stage it is instructive to look again at the relationship between $|\vec{l}|$ and t. Nyquist smoothing is applied again to this relationship to remove the jitter in the length

of the vector $|\vec{l}|$, and the increases in $|\vec{l}|$ at each peak can be clearly seen in the figure below:

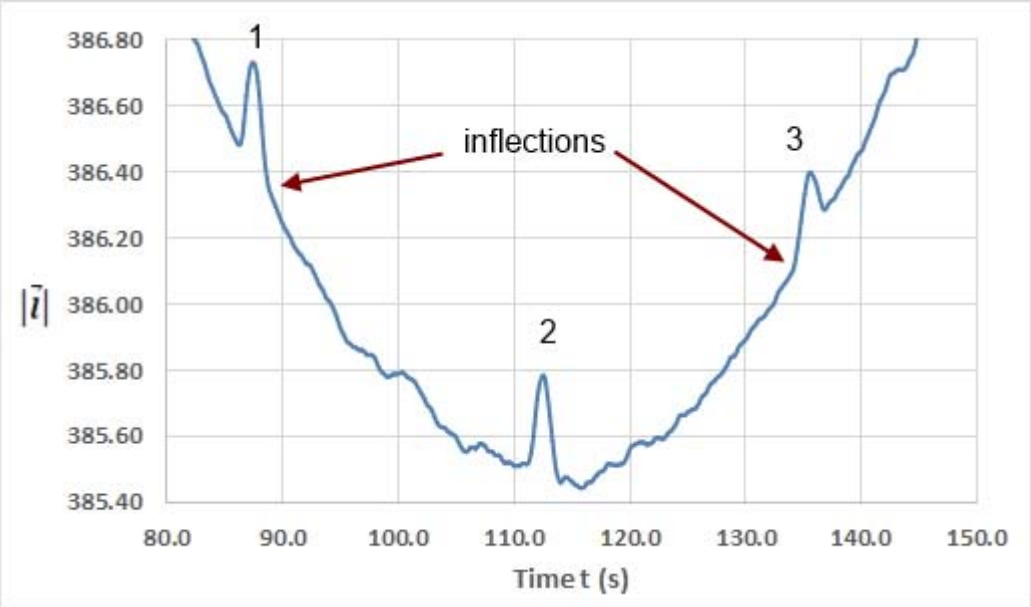


Figure 5.3.4: The electropherogram of figure 5.3.3 above is used to give the plot of $|\vec{l}|$~t over the data extract from 80.0 s to 150.0 s. The function is intrinsically smoothed with Nyquist to remove the jitter in $|\vec{l}|$ due to the high-noise environment. This does not significantly affect the positions of the critical points (t_{left} , A_{left}) , (t_{max} , A_{max}) and (t_{right} , A_{right}).

From the perspective of a visual examination to determine peak width, Figure 5.3.4 above turns out to be of little more use than the initial electropherogram of figure 5.3.3. The points (t_{left} , A_{left}) and (t_{right} , A_{right}) are points of inflection in both of these figures; that is, they are points where the curvature of the function changes from concave upward to concave downward or vice versa. The inflections highlighted in Figure 5.3.4 are very difficult to discern by visual inspection, but far simpler if one looks at the function $\frac{d|\vec{l}|}{dt}$ ~ t. The advantage of doing this is that a mean Value Theorem insertion of an extra column in a spreadsheet is very simple, and shows exactly where the tangential direction moves from a concave upward curve to a concave downward curve.

The data-based functions such as VLOOKUP, INDEX, MAX and MIN in Microsoft Excel™ can then be applied to the extracted dataset for unambiguous determination of peak position, baseline, peak height and peak width by identifying precise positions of the inflection points (t_{left} , A_{left}) , (t_{max} , A_{max}) and (t_{right} , A_{right}) from the functions.

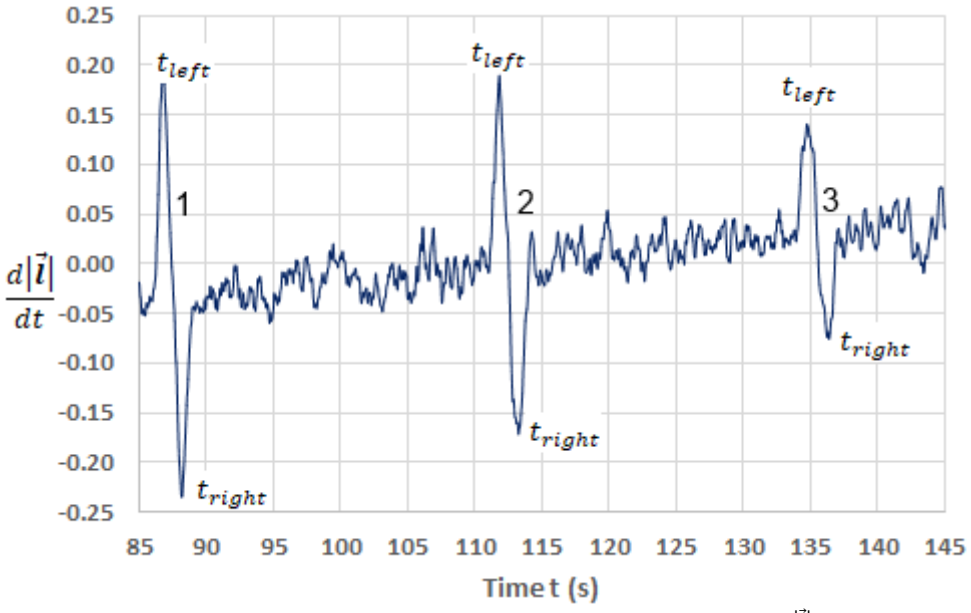


Figure 5.3.5: The maxima and minima for each peak on the function $\frac{d|\vec{l}|}{dt} \sim t$ show precisely the inflection points where curvature changes, and give an unambiguous position for (t_{left} , A_{left}) and (t_{right} , A_{right}) for each of the three signals in this dataset. This minimises unquantifiable error inherent in visual inspection, and allows more reliable determination of signal baseline.

If there is an error in visually determined baseline, then admittedly such an error is halved if one deals with peak height or peak area at half height. Perhaps the method described here may provide a better alternative.

This method allows in turn a detailed comparative analysis of performance of the composite algorithms by using the procedures outlined in Chapter 3 and explained further in Chapter 4.

5.3.1 Signal-to Noise Ratio

For determination of signal-to-noise ratio, firstly an average baseline, peak height and neighbouring noise are established from the illustrations above as follows:

- Vector method described above is used to determine (t_{left} , A_{left}) , (t_{max} , A_{max}) and (t_{right} , A_{right}).
- Baseline is determined from:

$$B = \frac{A_{(left)} + A_{(right)}}{2} \tag{5.3.1.1}$$

and peak height from:

$$H = A_{(max)} - B \tag{5.3.1.2}$$

- Noise segment using a window of 6000 points (~1.0 s) on either side of (t_{left} , A_{left}) and (t_{right} , A_{right}) is used and the two-dimensional standard deviation σ_{At} is calculated using the STEYX function.

- Total noise is then

$$\sigma_{At(total)} = \frac{\sigma_{At(left)} + \sigma_{At(right)}}{2} \tag{5.3.1.3}$$

- signal-to-noise ratio is then determined from:

$$SNR = \frac{H}{\sigma_{At(total)}} \tag{5.3.1.4}$$

5.3.2 Peak Widths

The peak width for each peak is simply determined as usual from:

$$W_p = t_{right} - t_{left} \tag{5.3.2.1}$$

where the terms t_{right} and t_{left} are shown in Figure 5.3.5.

Both SNR and peak width are derived directly from formulas embedded in the spreadsheets. Such formulas can be seen in APPENDIX 5.

5.3.3 Results

At this stage, it is instructive to look at comparative electropherograms of the 2 best performing pairs of nested algorithms. These are shown in figures 5.3.3.1 and 5.3.3.2 below. A visual inspection will show some interesting comparisons.

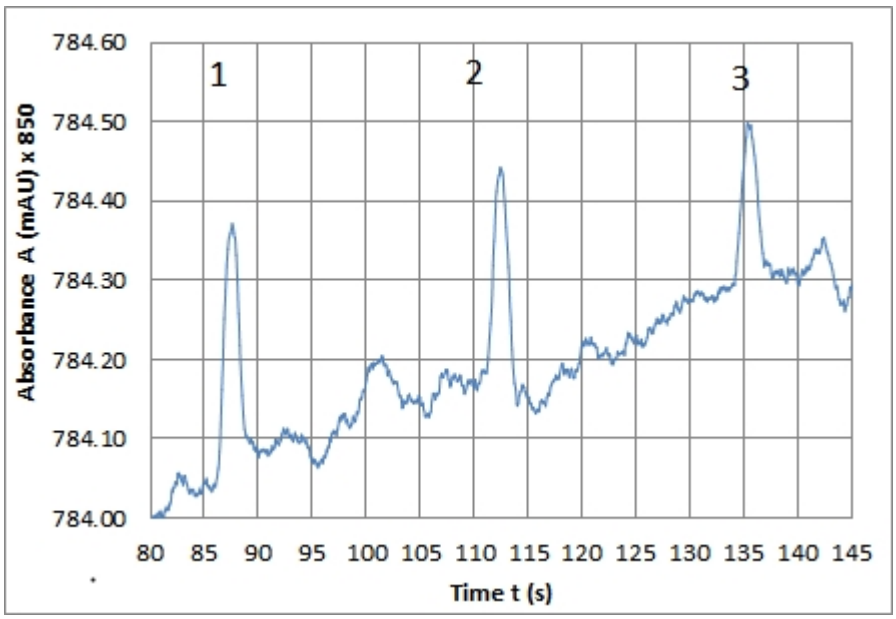


Figure 5.3.3.1: Electropherogram of the optimal SNR value of SG605(N241) pair of nested iterative algorithms. At these window values, $SNR_{peak\ 3} = 27.12$, which is 6.51 times the SNR_{peak3} for SG605 alone.

Although this electropherogram gives the maximum SNR for all 3 peaks, two things need to be noted:

1. there is a slight rounding of peaks when compared with the starting point which was Nyquist alone with 131 point window, giving an initial value for $SNR_{peak3} = 3.02$ shown in figure 5.3.3 above; and
2. the peak widening in Figure 5.3.3.1 needs to be noted for comparison with Figure 5.3.3.2 below.

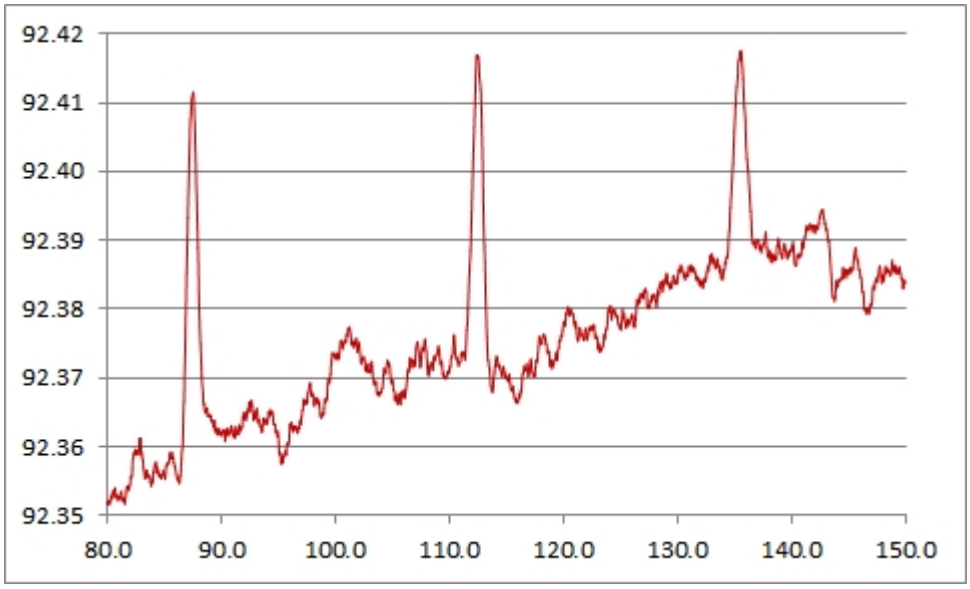


Figure 5.3.3.2: Electropherogram of the optimal SNR value of Gauss381(Nyq241) pair of nested iterative algorithms. At these window values, $SNR_{peak\ 3} = 25.03$, which is 6.10 times the SNR_{peak3} for Nyq(241) alone.

On this sample set, there is not much to choose between Nyquist nested inside Savitsky-Golay, and Nyquist nested inside Gauss. Visual inspection of Figure 5.3.3.1 with Figure 5.3.3.2 (taking into account the differences in scale) shows that the latter seems to be a better preserver of peak width, and this is borne out by the numbers shown in Table 5.3.4.1 below.

All peak window sizes were linked to a meta-analysis spreadsheet in the same workbook in which the critical data values for each pair of nested algorithms was calculated, using the methods outlined in Chapter 4 and in §§5.1-5.3 of Chapter 5.

For further visual inspection of changes in electropherogram characteristics as the internal algorithm iterates through window sizes as they increase by 10 points at a time, it is instructive at this juncture to refer to Figure Apx 5.3 in APPENDIX 5. It endeavours to illustrate the unique identification of peak turning point, irrespective of window smoothing size, and also the increase in peak width with window size.

This allowed successive calculation of Wp and SNR as each successive spreadsheet was calculated for each window size. Bearing in mind the relationship shown in Figure 5.2.1.6, there is a striking visual parallel to Figure 5.3.3.1 which follows. The relationships for Wp~n and SNR~n are shown below, and the datasets for these relationships can be seen in APPENDIX 5.

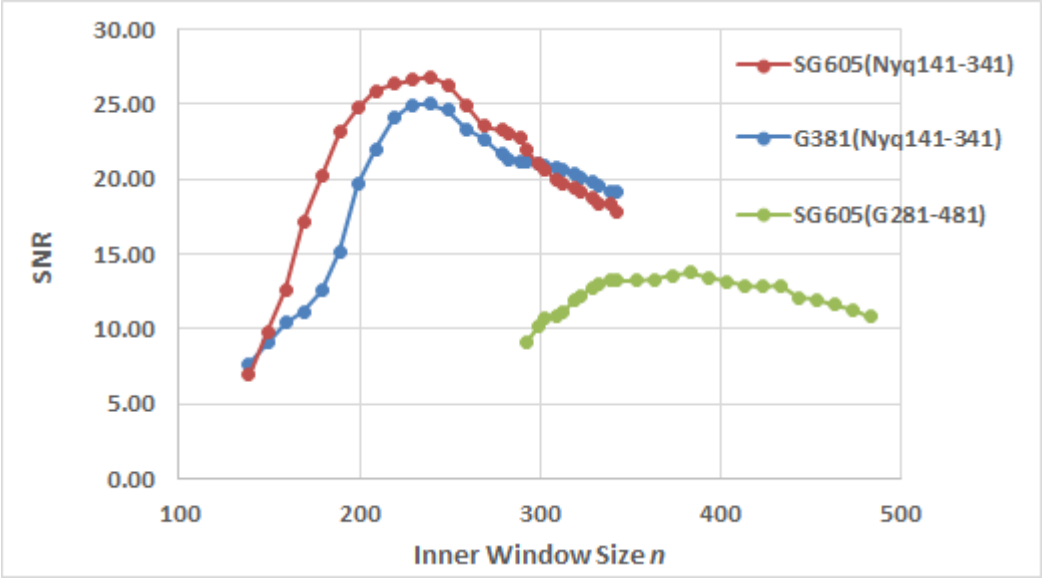


Figure 5.3.3.3: Second illustration of the relationship between SNR and three pairs of nested smoothing algorithms. Again the smaller internal algorithm window is iterated successively at 10-point width increments as shown in Table 5.1.2.1.1inside the fixed outer optimal window sizes. It turns out that the order of performance of the algorithms is the same as that which appears in the previous 3 kHz dataset.

As far as peak width is concerned, it may help to re-examine the Figure Apx. 3 in APPENDIX 5 to align the changes in peak width with the size of the internal smoothing window illustrated below in Figure 5.3.3.4.

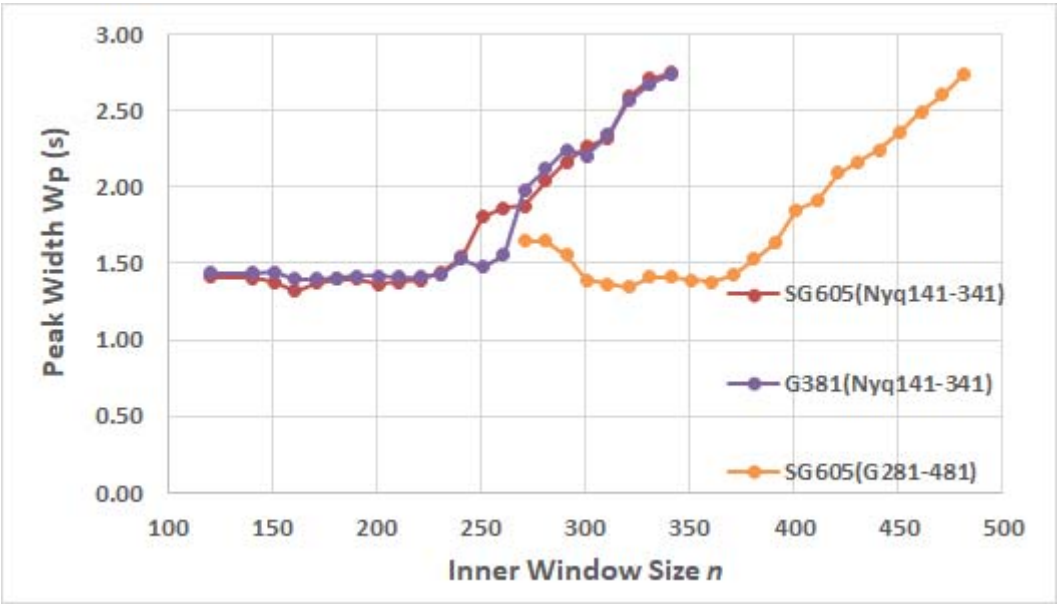


Figure 5.3.3.4: Comparitive performance of the three pairs of nested algorithms with respect to peak width.

On visual inspection of all three cases, the peak width begins to increase significantly only for window size greater than the optimum performance smoothing window size of the internal algorithm of each pair. That is for SG(N) and G(N) after 241 points and SG(G) after 381 points. The datasets can be seen in APPENDIX 5.

5.3.4 Performance Analysis and Ranking

Both unknown datasets, namely 3 kHz at 22 bits and 6 kHz at 16 bits contained three identifiable peaks. Neither dataset required analysis of resolution because all 3 peaks were clearly separated. The same analysis was applied to both datasets and the summary results are presented below.

Table 5.3.4.1: The results of peak width Wp, and SNR case of two unseen multiple-peak electropherograms run high DAQ. Again, as the in the case of the composite algorithms explored in Chapter 3, the factor by which the composite SNR is an improvement on the better of the values in the two individual algorithms is shown; also shown is the factor by which the peak width Wp of the compound algorithm differs from the smaller of the peak widths of its component algorithms.

	NYQ (241 pt)	**GAUSS (381 pt)**	**S-G (605 pt)**	**NYQ (241 pt) In S-G (605 pt)**	**NYQ (241 pt) In GAUSS (381 pt)**	**GAUSS (381 pt) In S-G (605 pt)**
3 kHz 22-bit SNR (factor of larger SNR)	4.40	6.22	5.49	36.07 (6.57)	27.30 (6.21)	21.83 (3.51)
3 kHz 22-bit Wp (factor of smaller Wp)	1.963 Max 3.896 @341 pts	2.011 Max 2.968 @481 pts	2.907 Max 2.907 @605 pts	1.696 (0.864)	1.802 (0.918)	1.929 (0.959)
6 kHz 16-bit SNR (factor of larger SNR)	4.10	3.97	4.22	27.12 (6.51)	25.03 (6.10)	14.71 (3.49)
6 kHz 16-bit Wp pk (factor of smaller Wp)	1.701 Max 2.736 @341 pts	1.683 Max 2.727 @481 pts	2.801 Max 2.801 @605 pts	1.527 (0.898)	1.552 (0.912)	1.609 (0.956)

Firstly, the compound algorithms perform on the two criteria of SNR and Wp in the same order for these two datasets, both of which were acquired at very high DAQ.

Secondly, there is good agreement (RSD<2% for both Wp and SNR) in the performance on each criterion when measured by the factors by which SNR is improved, and peak width is preserved.

5.4 Conclusions

It would seem that in Chapter 5, the application of the principles outlined in chapter 4 and the compound nested algorithms derived there are applicable to the given two fast DAQ/high noise datasets provided.

It is important to realise that when dealing with two nested algorithms of different mathematical structure, they will affect both signals and noise in different ways. This chapter has been an attempt to identify pairs of algorithms which will give optimum benefit by identifying the strengths and weaknesses of each. Judicious combination of algorithms allow for maximising one or more criteria of interest.

The use of a vector length with anchor point chosen carefully allows for the highlighting and unambiguous determination of signal turning points.

It would also seem as if this chapter indicates a possible proof of concept for such an approach to minimise signal distortion whilst at the same time providing a mathematical framework for identifying random noise and minimising its effect, whilst increasing SNR and enabling a focus on peak width at baseline. Removal of noise with minimal distortion allows for a clearer identification of signal turning points and hence a reliable baseline whose exact level might otherwise be obscured.

5.5 References

1. Shackman, J. G., Watson, C.J., Kennedy, R.T., High-throughput automated post-processing of separation data. *Journal of Chromatography A* **2004,** *1040* (2), 273-282.

2. Gao, L., Pulido, J.S., Hatfield, R.M., Dundervill, R.F., McCannel, C.A., Shippy, S.A., Capillary electrophoretic assay for nitrate levels in the vitreous of proliferative diabetic retinopathy. *Journal of Chromatography B* **2007,** *847* (2), 300–304.

3. Reiter, L., Rinner, O., Picotti, P., Huttenhain, R., Beck, M., Brusniak, M. Y., Hengartner, M. O., Aebersold, R., mProphet: Automated data processing and statistical validation for large-scale SRM experiments. *Nat Methods* **2011,** *8* (5), 430-435.

4. Kaigala, G. V., Hoang, V.N., Stickel, A., Lauzon, J., Manage, D., Pilarski, L.M., Backhouse, C.J., An inexpensive and portable microchip-based platform for integrated RT–PCR and capillary electrophoresis. *Analyst* **2008,** *133* (3), 331-338.

5. Laude, N. D., Atcherley, C.W., Heien, M.L., Rethinking Data Collection and Signal Processing – Real Time Oversampling Filter for Chemical Measurements. *Anal. Chem.* **2012,** *84* (19), 8422−8426.

6. Chacron, M. J., Lindner, B., Longtin, A., Noise shaping by interval correlations increases information transfer. *Phys Rev Lett* **2004,** *92* (8), 0806011-0806014.

7. Hsu, C.-M., Straayer, M.Z., Perrott, M.H., A Low-Noise Wide-BW 3.6-GHz Digital Fractional-N Frequency Synthesizer with a Noise-Shaping Time-to-Digital Converter and Quantization Noise Cancellation. *IEEE Journal of Solid-State Circuits* **2008,** *43* (12), 2776-2786.

8. Reymann, J., Baddeley, D., Gunkel, M., Lemmer, P., Stadter, W., Jegou, T., Rippe, K., Cremer, C., Birk, U., High-precision structural analysis of subnuclear complexes in fixed and live cells via spatially modulated illumination (SMI) microscopy. *Chromosome Research* **2008,** *1* (16), 367-382.

9. Innocente, V., Silvestris, L., Stickland, D., CMS software architecture - Software framework, services and persistency in high level trigger, reconstruction and analysis. *Computer Physics Communications* **2001,** *140*, 31-44.

10. Felinger, A., Guiochon, G., Validation of chromatography data analysis software. *Journal of Chromatography A* **2001,** *913*, 221-231.

11. Gallo, C., Capozzi, V., Lasalvia, M., Perna, G., An algorithm for estimation of background signal of Raman spectra from biological cell samples using polynomial functions of different degrees. *Vibrational Spectroscopy* **2016,** *83*, 132-137.

12. Rakotomamonjy, A., Surveying and comparing simultaneous sparse approximation (or group-lasso) algorithms. *Signal Processing* **2011,** *91*, 1505-1526.

6 Concluding Summary and Proposed Future Directions

6.1 Summary

To conclude this work, it is prudent to revisit its beginning. It began as a project to construct a miniaturised CE device using cheap, off-the-shelf components, with self-designed software, fabricated connections and ancillary components. During the construction and testing of this miniaturised instrument, much was learned about the difficulties implicit in the management of injection times, flow control, high-voltage insulation, fast migration time and short capillaries. Nonetheless, an independently powered functioning instrument was created and the difficulties themselves proved to be informative for future development.

In writing the control software for the instrument, C++ was found to be a reliable and efficient language and there seems no reason to change this at the present time. At this point the distinction must be made between the control software for the instrument, and the data processing software used in analysis. The latter had its shortcomings which will be discussed in §6.3.

From the very rough digitised data obtained using cheap 10 bit ADC, sprang the fortuitous change in direction of this study. This led to a study of smoothing algorithms and their performance relative to a novel application of the Nyquist Theorem, which had hitherto not been used in this particular way in signal analysis in standard CE or microfluidics. This led to a search for improvement in SNR and signal characteristics by using pairs of algorithms which were iterative and nested. When these nested algorithms were applied to a high noise, very low SNR environment, significant improvements were achieved.

6.2 Directions in Construction

It would seem from this study that much effort in the research community is being spent in the search for a "παγκόσμια αναλυτική μηχανή" (pancosmic – an all-purpose) miniaturised analytical machine. The rise of Caliper Technologies led to acquisitions of companies which were able to broaden the repertoire of ancillary technology in order to widen the scope of miniaturised CE and related microfluidic devices[1]. The lesson which may be drawn from the subsequent fall of Caliper Technologies could be that a large base of

useful technology seemed not to lead to consolidation and integration of hardware, but rather to broaden directions to an extent where initial strengths and focus could be lost.[2]

The ubiquity and decreasing cost of 3-D printing has led to a great deal of excitement in the microfluidics community, and with some justification, because the technology is both new and exciting and seems to have the potential for personalised and custom design of small instruments which can be endlessly tested and modified. The danger with new technology – as was the case with the initial use of the desktop computer and latterly with the laptop – is that the expectations often outrun the capability[3] of the technology.

The direction decided after this study is to pursue a miniaturised CE device with improved functionality by replacing the Arduino with Rasberry Pi.[4] This will allow improved functionality, foster DAQ and higher speed. The Rasberry Pi chip is programmable and does not require the insertion of separate software to convert the digital data to a .csv file.

Further, this means the construction of a single-purpose or limited purpose device, most specifically to address the problem of lead in drinking water in places such as Flint, Michigan[5], Port Pirie, South Australia and Mount Isa[6], Queensland.

It is envisaged that such a device will be designed and modified using 3-D printing, where 3-D printing will replace moulding, and an integrated container will be custom designed to hold pumps, solenoids, capillary, microchip, and battery[7]. Having said this, however, it is true that 3-D printing is latterly being used to create ever-more ingenious components[8] which enhance efficiency and provide new functionality in miniaturised analytical instruments. Only by using a cautious and limited approach can the temptation for an early and unrealistic expectation be minimised.

6.3 Directions in Software and Processing Speed

The 2018 processing speed of the fastest Rasberry Pi is 1.2 GHz, with 64-bit quad core processor, compared to Arduino Uno's 16M Hz and 10-bit chip. This makes a significant difference and will allow primary programming for data acquisition and data to be put directly into a suitable spreadsheet format. It is envisaged that this date are will then be transferred via a separate port to a storage chip for post-facto processing. During this study, post facto processing was carried out using Microsoft Excel™, and the large volumes of data ($\sim 10^6$ datum points) meant that time taken each smoothing process on

a spreadsheet window could be as long as 15 minutes. Three different computers were used, all of which had 16 GB of RAM, and processors with speeds of 1.8 GHz, 2.4 GHz, and 3.3 GHz. It was not really possible to make a meaningful comparison because memory and processing is also occupied by whatever happens to be running in the system tray at any time. A comparison was able to be made on the 3.3 GHz machine between Microsoft Excel™ and the open source software Libre Office Calculator. The latter software proved to be about 20% faster, but a more systematic study of this possibility will need to be made.

A further improvement in processing time could be made by a preliminary software rough detection of signals, and thereafter extracting only the segment of interest for analysis from the dataset. This process was carried out manually with an initial smoothing shown in Figure 5.3.2. It would not be a difficult matter for a software loop to detect the 3 signals shown and extract them together with a small window on each end to enable calculation of SNR. In high DAQ/high noise data, embedded computational analysis is very time expensive, and extraction of such a segment can significantly reduce computing time.

A further issue which arose was the notion that variations in the maximised SNR when using different datasets at different DAQ would suggest that there might be a relationship between SNR maximum in the smoothing algorithm (whether single or nested), and that this may be related to peak width or some other time-dependent relationship. Such an investigation may also offer some further insights into any relationship between peak characteristics and DAQ.

6.4 Directions in Algorithm Programming

In this study, both the inner and outer algorithms are iterative[9]. Although the outer algorithm may be mathematically allowable as recursive, this possibility is left for future study.

The question now arises as to whether this method compromises peak shape to any extent, and so possibly compromises a reliable way of integrating to find peak area. Integration by post-facto computation is most reliable when one uses the trapezoidal rule of ***inf(sup) – sup(inf)***, using greatest lower bound and least upper bound respectively -particularly when DAQ is very high. This approach to continuity as discussed in Chapters 3 and 4 makes the finding of each limit a matter of a single line of

code, and the area is unambiguous when the baseline turning points are determined by using the second derivative of the length vector $\frac{d|\vec{l}|}{dt} \sim t$ as in Chapter 5.

In the work of Dubský, Dvořak et.al,[10] any compromise in peak shape would also compromise the determination of migration time from the Haarhoff–van der Linde (HVL) function and its application to peak geometry after the method of Erny, et.al. [11].

It is envisaged that future development of the instrumental operating system might incorporate a subroutine to describe non-Gaussian peaks with the Haarhoff-van der Linde equation and so enhance the accuracy of peak areas and migration times. There is also the Kalembet method of combining EMS and Gauss.[12] and this needs to be explored, together with other proportional combinations of nested algorithms to compare such combinations with HVL.

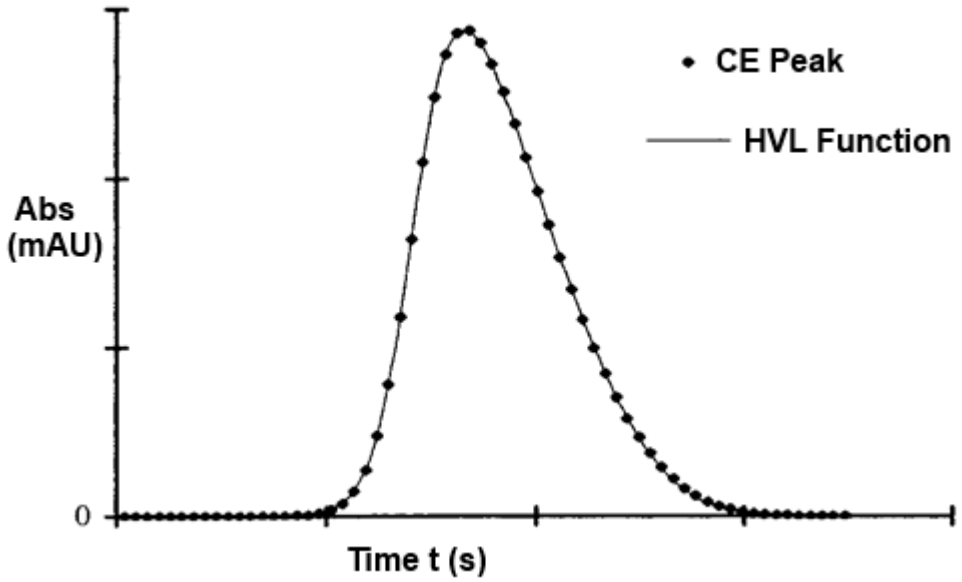


Figure 6.4.1: A non-Gaussian peak approximated by simulation of an HVL function. This could realistically be improved by using the method of Kalembet, or a proportional combination of nested algorithms.

The HVL equation is potentially very useful, and able to model exactly the non-Gaussian types of CE signals (curved increase, vertical decrease; vertical increase, curved drop) which frequently occur. However, there is an intuitive case for comparing the HVL approximation with a proportional description using the weighted nesting of two algorithms as a means of achieving the closest mathematical description as an approximation to any given signal.

What would be the test criterion? The answer must lie in the trapezoidal rule for area. The original signal obtained with high DAQ will be able to provide an accurate computational outcome for area based on the original dataset.

6.5 References

1. Volpatti, L. R., Yetisen, A. K., Commercialization of microfluidic devices. *Trends Biotechnol* **2014,** *32* (7), 347-350.

2. Rios, A., Zougagh, M., Avila, M., Miniaturization through lab-on-a-chip: utopia or reality for routine laboratories? A review. *Anal Chim Acta* **2012,** *740*, 1-11.

3. Sharma, S.; Plistil, A.; Simpson, R. S.; Liu, K.; Farnsworth, P. B.; Stearns, S. D.; Lee, M. L., Instrumentation for hand-portable liquid chromatography. *J Chromatogr A* **2014,** *1327*, 80-9.

4. Urban, P. L., Open-Source Electronics As a Technological Aid in Chemical Education. *Journal of Chemical Education* **2014,** *91* (5), 751-752.

5. Tang, X., Wang, P-Y., Buchter, G., Ion-Selective Electrodes for Detection of Lead (II) in Drinking Water: A Mini-Review. *Environments* **2018,** *5* (9), 1-14.

6. Mackay, A. K., Taylor, M. P., Munksgaard, N. C., Hudson-Edwards, K. A., Burn-Nunes, L., Identification of environmental lead sources and pathways in a mining and smelting town: Mount Isa, Australia. *Environ Pollut* **2013,** *180*, 304-311.

7. Koenka, I. J., Saiz, J., Rempel, P., Hauser, P. C., Microfluidic Breadboard Approach to Capillary Electrophoresis. *Anal Chem* **2016,** *88* (7), 3761-3767.

8. Adamopoulou, T., Deridder, S., Desmet, G., Schoenmakers, P. J., Two-dimensional insertable separation tool (TWIST) for flow confinement in spatial separations. *Journal of Chromatography A* **2018**.

9. Bangliang, S., Yiheng, Z., Lihui, P., Danya, Y., Baofen, Z., The use of simultaneous iterative reconstruction technique for electrical capacitance tomography. *Chemical Engineering Journal* **2000,** *77*, 37-41.

10. Dubsky, P., Dvorak, M., Mullerova, L., Gas, B., Determination of the correct migration time and other parameters of the Haarhoff-van der Linde function from the peak geometry characteristics. *Electrophoresis* **2015,** *36* (5), 655-661.

11. Erny, G. L.; Bergström, E. T.; Goodall, D. M.; Grieb, S., Predicting Peak Shape in Capillary Zone Electrophoresis: a Generic Approach to Parametrizing Peaks Using the Haarhoff−Van der Linde (HVL) Function. *Analytical Chemistry* **2001,** *73* (20), 4862-4872.

12. Kalambet, Y., Kozmin, Y., Mikhailova, K., Nagaev, I., Tikhonov, P., Reconstruction of chromatographic peaks using the exponentially modified Gaussian function. *Journal of Chemometrics* **2011,** *25* (7), 352-356.

APPENDIX 1

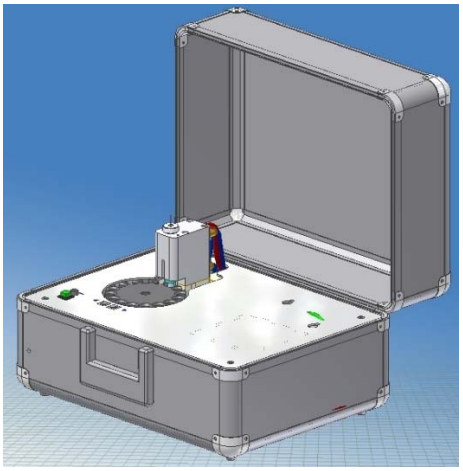


Figure Apx 1.1: CEP-5100 Autosampler by EH Systems™ (2003) which is a portable sampler using a rotating carousel for sample vial insertion. It requires the external coupling of a separate data acquisition system.

Figure Apx 1.2: The low-cost apparatus described by Marini et.al.,(2010) for the verification of pharmaceutical products and detection of counterfeit medicines. The instrument is portable, but requires dismantling and re-assembly.

Figure Apx 1.3: The interior of the triple-channel portable CE instrument of Mai, Hauser, et.al (2015) It has a reported weight of 15.0 kg with 8 h of battery operation for one channel, but only 2.5 h if all three channels are used. It uses pneumatic pressure and a complex system of pressure sensors.

Figure Apx 1.4: The handheld ITP instrument (7.6 x 5.7 x 3.8 cm) reported by Kaigala, Bercovici, Benham, Elliott and Santiago (2016) powered from a USB link and a laptop computer. The instrument is reported to be self-contained and includes a 5 mW laser, photodiode, high-voltage generation, switching, and data communication chips. The casing is of metal and is reported to act as a Faraday cage. The cost is not reported. This brief report was accessed in September 2018 at http://techmicro.siteexpress.co.il/research/itp-on-a-hand-held-device/

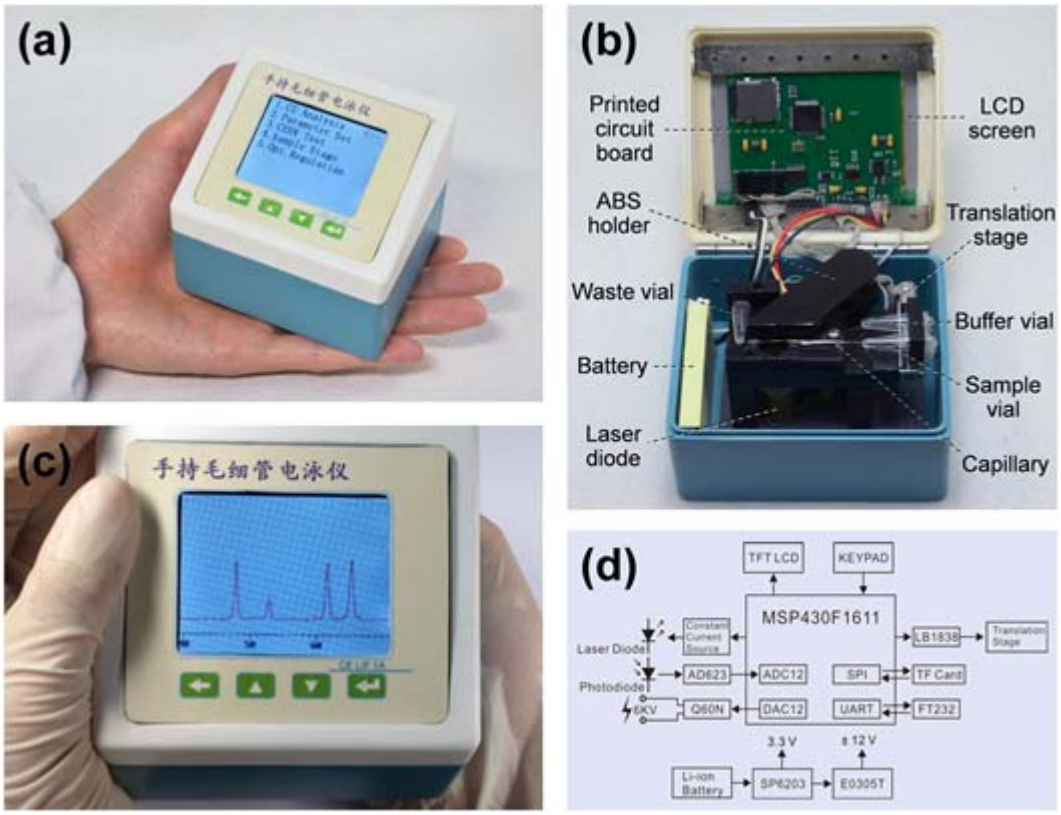


Figure Apx 1.5: Low-cost (≅ $500) palmtop high-speed CE Bioanalyzer with laser-Induced fluorescence detection reported by Pan, Fang, Hu (2018)

(a) Overall appearance;

(b) Uncovered appearance;

(c) An electropherogram of separation of amino acids (FITC-labeled arginine, phenylalanine and glycine) is displayed on the screen in real time;

(d) Schematic diagram of total electronic module of the bioanalyzer.

APPENDIX 2

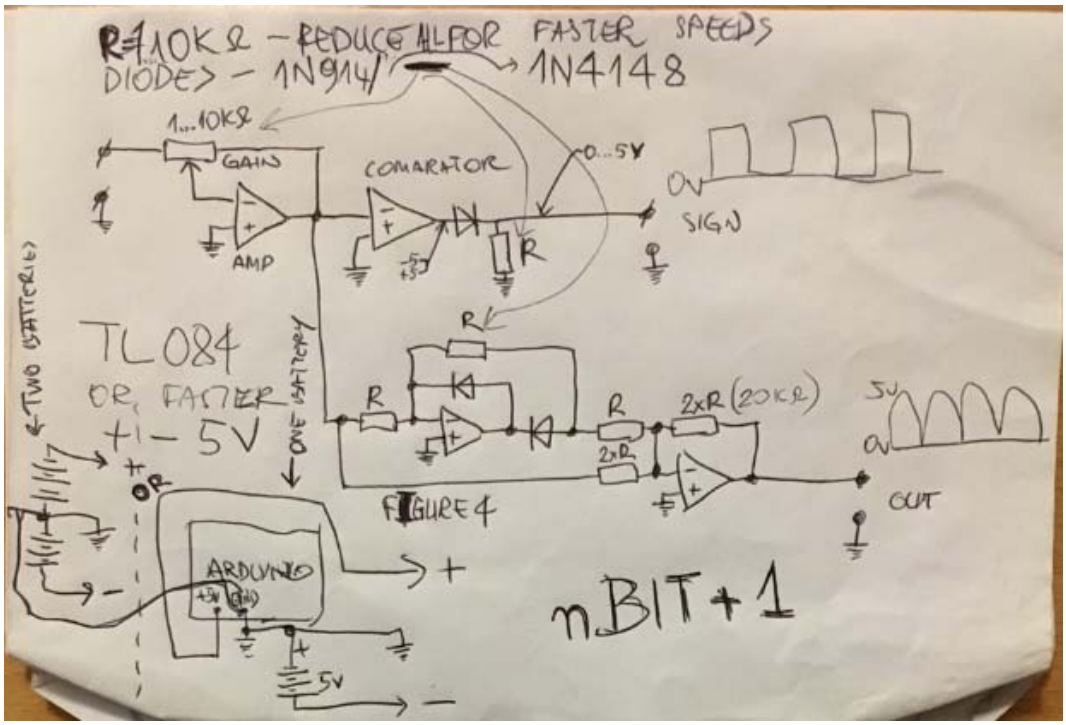


Figure Apx 2.1: An ingenious method for increasing the resolution of a 10-bit Arduino UNO to 11 bits. The hacker states (incorrectly) that this will double the resolution. Resolution is improved without significant loss of detail provided the DAQ is low. For fast or very fast DAQ (see definitions) there is a lag time introduced by the parallel diodes. This configuration is data-intrusive. https://hackaday.com/2018/05/07/double-the-resolution-from-an-arduino-adc/

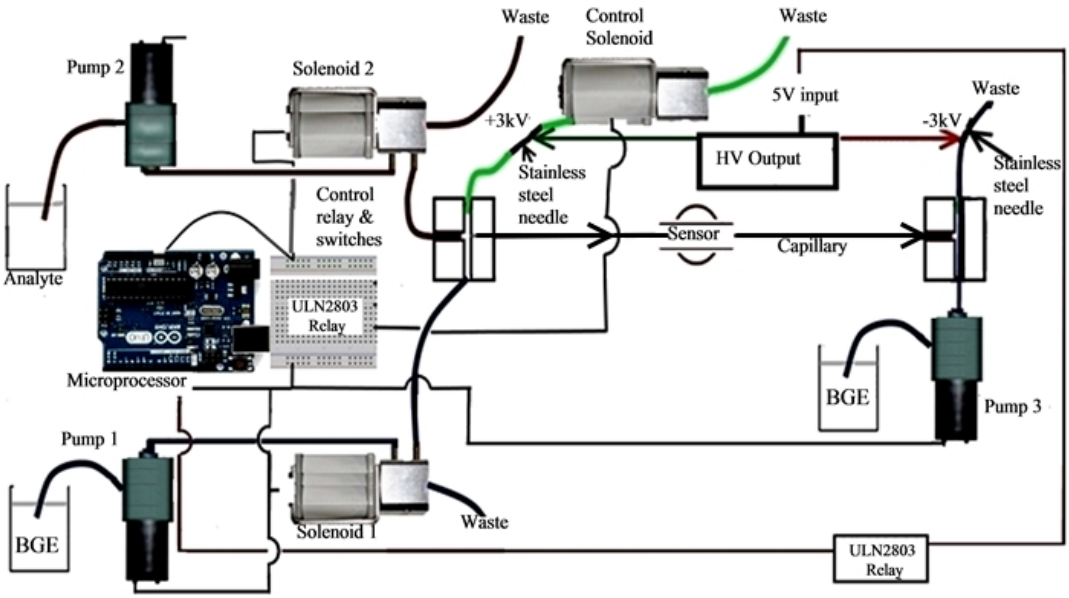


Figure Apx 2.2: More detailed schematic of the instrument described in Chapter 2 §2.1.3 Figure 2.1.3.1 showing actual components.

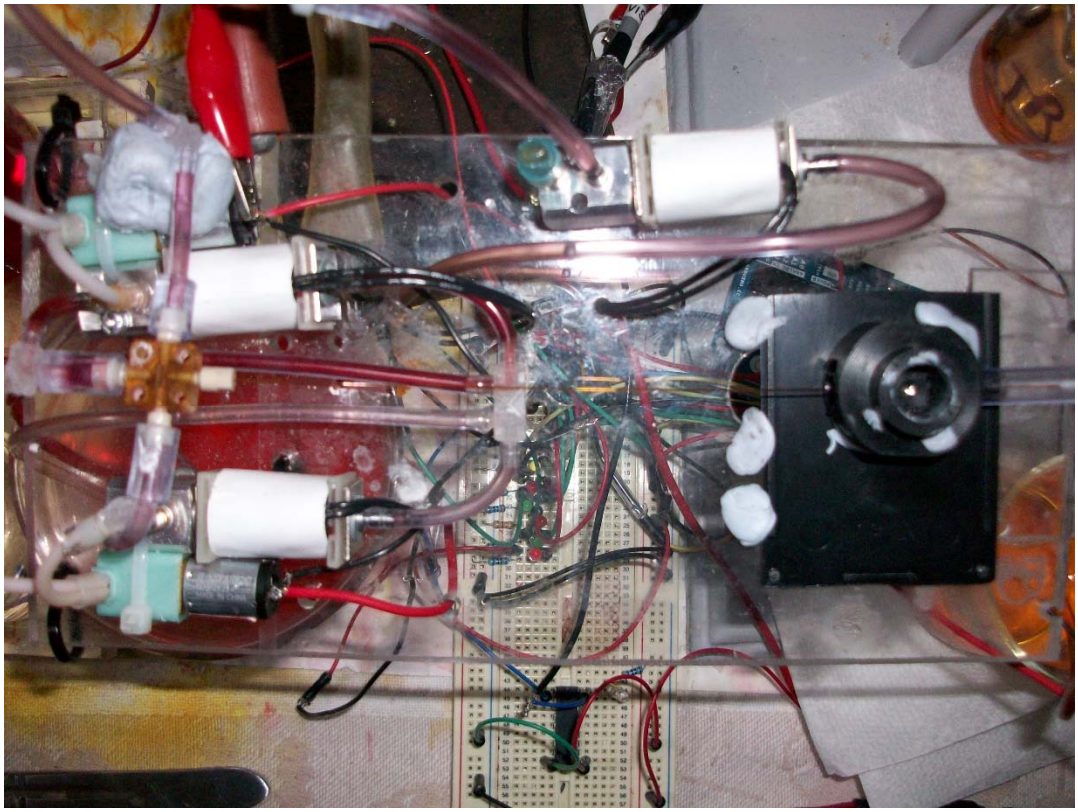


Figure Apx 2.3: Photograph of the instrument described in Chapter 2 during operation. The pumps and solenoids can be clearly identified - this photograph needs to be read in conjunction with Figure Apx 2.2 above.

OPERATIONAL OUTLINE

The miniaturised CE apparatus described in Chapter 2 is automatically controlled in repeated 12-step cycles by a C++ program controlled by 10 bit ADC on Arduino Uno.

```
/*STEP 0: INITIALISE MICROCHIP PINS AND VARIABLE NAMES
      INITIALISE INTEGER VARIABLES FOR APPARATUS CONTROL;
      AND INITIALISE VARIABLES FOR INPUTS AND OUTPUTS*/

/* STEP 1: INITIALISE CHIP TO STABILISE INPUT AND OUTPUT PORTS

/*STEP 2: PRE-CONDITIONING AND PRIMING THE PUMPS AND LINES*/
/*All three pumps switch on and flush for "PumpsPrime" secs
from inlet to waste. This allows time for the pumps to prime,
and eliminates bubbles in tubing.*/

/*STEP 3. FLUSH BUFFER THROUGH CROSSPIECE
 AND CONTROL SOLENOID TO WASTE*/
 /*Turn on buffer solenoid for "Bufferflushcross" seconds to flush.
 This flushes buffer through the crosspiece and past capillary entrance
 for "Bufferflushcross" seconds*/

/*STEP 4. FLUSH BUFFER THROUGH CAPILLARY*/
 /*Turn on the Control Solenoid which will flush
 buffer through the capillary for "Bufferflushcap" seconds*/

/* STEP 5. OPEN CONTROL SOLENOID BRIEFLY TO RELEASE BUFFER
 PRESSURE AND PREVENT BACKFLOW.*/

/* STEP 6. OPEN BUFFER SOLENOID TO STOP FLOW THROUGH CAPILLARY.*/

/* STEP 7.CLOSE ANALYTE SOLENOID AND OPEN CONTROL SOLENOID;
  THIS FLUSHES ANALYTE THROUGH THE CROSSPIECE TO WASTE*/
```

/*STEP 8. INJECTION: CLOSE OFF CONTROL SOLENOID
TO INJECT ANALYTE PLUG INTO CAPILLARY.*/

/*STEP 9. OPEN CONTROL SOLENOID BRIEFLY TO RELEASE ANALYTE
PRESSURE AND PREVENT BACKFLOW.*/

/*STEP 10. ISOLATE THE PLUG BY FLUSHING BUFFER THROGUH
THE CROSSPIECE TO WASTE.
ALSO TURN ON SENSOR*/

/*STEP 11. DRIVE THE ANALYTE PLUG INTO THE CAPLILLARY
BY FORCING BUFFER BEHIND IT INTO THE CAPILLARY FOR 5 SEC.*/
//TURN ON HV /START READING

/*STEP 12: RE-SET PARAMETERS AND RETURN TO
THE "do" loop JUST BEFORE STEP 3. */

//FINISH

The entire program is listed below.

```
//PROGRAM TO CONTROL CE APPARATUS IN 12 STEPS
//HYDRODYNAMIC INJECTION AND ELECTROPHORESIS

/*STEP 0: INITIALISE MICROCHIP PINS AND VARIABLE NAMES
      INITIALISE INTEGER VARIABLES FOR APPARATUS CONTROL;
      AND INITIALISE VARIABLES FOR INPUTS AND OUTPUTS*/

//MICROCHIP PINS AND VARIABLE NAMES
```

```
int PumpBuffer = 13;
int SolenoidBuffer = 12;
int PumpAnalyte = 11;
int SolenoidAnalyte = 10;
int PumpEndCap = 9;
int SolenoidCtrl = 8;
int HighVolt = 7;
int SensorLight = 6;
int Photocell = 5;

//Initialise Variables for Input and Output Values
float PrecisionBits = 14.84;
int OversampleNum=822;
float SensePin = analogRead(A0);
float AnalyteSense = analogRead(A1);
float BufferSense = analogRead(A2);
int p2 = analogRead(A1);
int p1 = analogRead(A1);
int n=0;
int pwm=180;
void setup()
{
```

```
 //INITIALISE DATA READ INPUTS
Serial.begin(115200);
analogReference(DEFAULT);
 /*This stays at 5.0V to control injection, after which it
 reverts to 1.1V during data collection and downloading.*/

 // INITIALISE APPARATUS CONTROL OUTPUTS

 pinMode(PumpBuffer,OUTPUT);
 pinMode(SolenoidBuffer,OUTPUT);
 pinMode(PumpAnalyte,OUTPUT);
 pinMode(SolenoidAnalyte,OUTPUT);
 pinMode(PumpEndCap,OUTPUT);
 pinMode(SolenoidCtrl,OUTPUT);
 pinMode(HighVolt,OUTPUT);
 pinMode(SensorLight,OUTPUT);
}
void loop()
 {
   //Initialise all Time Controls and Chemical Variables.
   //String1 is BGE.
   //String2 is Analyte.
 float Capdiam = 25;
 float Caplength = 17.0;
 float Windowposition = 12.5;
 float LEDCurrent = 30.0;
 float Injectiontime = 8.0;
 float SepnVolt = 5920.0;
 char Str1[ ] = "Potassium dihydrogen Phosphate/PAR  " ;
 char Str2[ ] = "Zn/PAR" ;
 float BGEConc = 10.0;
```

```
float BGEpH = 6.9;
float AnalyteConc = 10.0;
int InitialiseChip = 4000;
int PumpsPrime = 10000;
float Bufferflushcross = 13000;
float Bufferflushcap = 45000;
float BufferPress = 500;
float Analyteflushcross = 13000;
float AnalytePress = 500;
float BufferClearCross = 13000;
float Plugtime = 2000;
float TimeSepn = 400000;
int ReadDelay = 100;
int Setnumber = 30;
int TimeBetweenSets = 20000;

//READY TO START
/* STEP 1: INITIALISE CHIP TO STABILISE INPUT AND OUTPUT PORTS
 /*Wait "InitialiseChip" seconds before commencement.
 Allows pre-setting of microchip and transistor relays.*/

 delay(InitialiseChip);
Serial.println("STEP 1 COMPLETE");

/*STEP 2: PRE-CONDITIONING AND PRIMING THE PUMPS AND LINES*/

/*All three pumps switch on and flush for "PumpsPrime" secs
from inlet to waste. This allows time for the pumps to prime,
and eliminates bubbles in tubing.*/

 digitalWrite(PumpBuffer,HIGH);
 analogWrite(PumpAnalyte,180);
```

```
digitalWrite(PumpEndCap,HIGH);
delay(PumpsPrime);
Serial.println("STEP 2 COMPLETE");
/* All three pumps are now left running.
This is where the run sequence begins.
Start to count the number of run sequences.
THE PROGRAM RETURNS TO THIS POINT FOR EACH RUN UNTIL
THE SET IS COMPLETE*/

int k = 0;
//THIS IS WHERE EACH NEW RUN STARTS FROM**
 do
{
k = k+1;

/*STEP 3. FLUSH BUFFER THROUGH CROSSPIECE
AND CONTROL SOLENOID TO WASTE*/

/*Turn on buffer solenoid for "Bufferflushcross" seconds to flush.
This flushes buffer through the crosspiece and past capillary entrance
for "Bufferflushcross" seconds*/

digitalWrite(SolenoidBuffer,HIGH);

delay(Bufferflushcross);
Serial.println("STEP 3 COMPLETE");

/*STEP 4. FLUSH BUFFER THROUGH CAPILLARY*/
/*Turn on the Control Solenoid which will flush
buffer through the capillary for "Bufferflushcap" seconds*/

digitalWrite(SolenoidCtrl,HIGH);
```

```
 delay(Bufferflushcap);
Serial.println("STEP 4 COMPLETE");

/* STEP 5. OPEN CONTROL SOLENOID BRIEFLY TO RELEASE BUFFER
 PRESSURE AND PREVENT BACKFLOW.*/

 digitalWrite(PumpBuffer,LOW);
 delay(1000);
 digitalWrite(SolenoidBuffer,LOW);
 delay(1000);
 digitalWrite(SolenoidBuffer,HIGH);
 digitalWrite(SolenoidCtrl,LOW);
 delay(BufferPress);

 Serial.println("STEP 5 COMPLETE");

/* STEP 6. OPEN BUFFER SOLENOID TO STOP FLOW THROUGH CAPILLARY.*/

  digitalWrite(SolenoidBuffer,LOW);
  delay(500);
  digitalWrite(PumpBuffer,HIGH);

 delay(1500);

 Serial.println("STEP 6 COMPLETE");

/* STEP 7.CLOSE ANALYTE SOLENOID AND OPEN CONTROL SOLENOID;
   THIS FLUSHES ANALYTE THROUGH THE CROSSPIECE TO WASTE*/

 /*Turn on analyte for "Analyteflushcross" seconds.
 This gets analyte through the crosspiece junction.
```

```
 Analyte will now be ready at capillary end of the crosspiece.*/

 analogWrite(PumpAnalyte,180);
 delay(1000);
 digitalWrite(SolenoidAnalyte,HIGH);
 digitalWrite(SolenoidCtrl,LOW);
 delay(Analyteflushcross);
analogWrite(PumpAnalyte,pwm);
delay(2000);
Serial.println("STEP 7 COMPLETE");

 /*STEP 8. INJECTION: CLOSE OFF CONTROL SOLENOID
   TO INJECT ANALYTE PLUG INTO CAPILLARY.*/
int n=0;
 do
{
  int p1=analogRead(A1);
  int p2=analogRead(A1);

  if(((p2-p1)>0)&&(p2>862)&&(p2<864))
  {
/* INJECT SAMPLE BY SHUTTING OFF CONTROL
SOLENOID FOR INJECTION TIME*/

 //digitalWrite(PumpBuffer,HIGH);
  //digitalWrite(SolenoidBuffer,LOW);
 //analogWrite(PumpAnalyte,pwm);
 digitalWrite(SolenoidAnalyte,HIGH);
 //digitalWrite(PumpEndCap,HIGH);
 digitalWrite(SolenoidCtrl,HIGH);
 //digitalWrite(HighVolt,LOW);
 //digitalWrite(SensorLight,LOW);
```

```
 delay(Injectiontime*1000);
 Serial.print ("p2 = ");
 Serial.print(p2);
 n=n+1;
 }else delay(0);

}while (n<1);

 Serial.println();

 Serial.print("INJECTED FOR ");
 Serial.print(Injectiontime);
 Serial.print(" SECONDS");
 Serial.println();
 Serial.println("STEP 8 COMPLETE");


/*STEP 9. OPEN CONTROL SOLENOID BRIEFLY TO RELEASE ANALYTE
 PRESSURE AND PREVENT BACKFLOW.*/

 //digitalWrite(PumpBuffer,HIGH);
 //digitalWrite(SolenoidBuffer,LOW);
  analogWrite(PumpAnalyte,0);
  delay(1000);
 //digitalWrite(SolenoidAnalyte,HIGH);
 //digitalWrite(PumpEndCap,HIGH);
 digitalWrite(SolenoidCtrl,LOW);
  //digitalWrite(HighVolt,LOW);
 //digitalWrite(SensorLight,LOW);

 delay(AnalytePress);
```

```
 analogWrite(PumpAnalyte,180);
 Serial.println("STEP 9 COMPLETE");

/*STEP 10. ISOLATE THE PLUG BY FLUSHING BUFFER THROGUH
 THE CROSSPIECE TO WASTE.
 ALSO TURN ON SENSOR*/

 //digitalWrite(PumpBuffer,HIGH);
 digitalWrite(SolenoidBuffer,HIGH);
 //analogWrite(PumpAnalyte,180);
 digitalWrite(SolenoidAnalyte,LOW);
 //digitalWrite(PumpEndCap,HIGH);
 digitalWrite(SolenoidCtrl,LOW);
 //digitalWrite(HighVolt,LOW);
 digitalWrite(SensorLight,HIGH);

 delay(BufferClearCross);
 Serial.println("STEP 10 COMPLETE");

/*STEP 11. DRIVE THE ANALYTE PLUG INTO THE CAPLILLARY
BY FORCING BUFFER BEHIND IT INTO THE CAPILLARY FOR 5 SEC.*/


 //digitalWrite(PumpBuffer,HIGH);
 //digitalWrite(SolenoidBuffer,HIGH);
 //analogWrite(PumpAnalyte,180);
 //digitalWrite(SolenoidAnalyte,LOW);
 //digitalWrite(PumpEndCap,HIGH);
 digitalWrite(SolenoidCtrl,HIGH);
 delay(Plugtime);
 digitalWrite(SolenoidCtrl,LOW);
 digitalWrite(HighVolt,HIGH);
```

```
 //digitalWrite(SensorLight,HIGH);


 Serial.println("STEP 11 COMPLETE");


 analogWrite(PumpAnalyte,0);
 delay(300);
 analogReference(INTERNAL);


//START READING


 Serial.print("START OF RUN NUMBER - ");Serial.print(k-1);
 Serial.print("  RESOLUTION = ");Serial.print(PrecisionBits);Serial.print("Bits ");
 Serial.print("   RUN TIME = ");Serial.print(TimeSepn/1000);Serial.print(" s");
 Serial.print("    SAMPLING RATE = ");Serial.print(1000/ReadDelay);Serial.print(" Hz
");
 Serial.print(" CAPILLARY DIAMETER = "); Serial.print(Capdiam);Serial.print("um  ");
 Serial.print("  CAPILLARY LENGTH = ");Serial.print(Caplength);Serial.print("cm ");
 Serial.println();
 Serial.print("     WINDOW POSITION =
");Serial.print(Windowposition);Serial.print("cm ");
 Serial.print("LED CURRENT = ");Serial.print(LEDCurrent);Serial.print(" mA");
 Serial.print("    INJECTION TIME = ");Serial.print(Injectiontime);Serial.print("s ");
 Serial.print("  SEPARATION VOLTAGE = ");Serial.print(SepnVolt);Serial.print(" V");
 Serial.print("    FIELD STRENGTH =
");Serial.print(SepnVolt/Caplength,1);Serial.print(" V/cm");
 Serial.println();
 Serial.print("BGE is: ");Serial.print(Str1);
 Serial.print("BGE Concentration = ");Serial.print(BGEConc);Serial.print(" mmol/L");
 Serial.print("     BGE pH = ");Serial.print(BGEpH);
 Serial.print("  Analyte is: ");Serial.print(Str2);
 Serial.print("     Analyte Concentration =
");Serial.print(AnalyteConc);Serial.print("ppm ");
 Serial.println();
```

```
 Serial.print("*APPARATUS CONTROL PARAMETERS*");

 Serial.println();

 Serial.print("STEP 1:: ");Serial.print("Initialise Chip:
");Serial.print(InitialiseChip/1000);Serial.print(" s ");

 Serial.print(" STEP 2:: ");Serial.print("Pumps prime:
");Serial.print(PumpsPrime/1000);Serial.print(" s");

 Serial.print("  STEP 3:: ");Serial.print("Buffer flushes Crosspiece:
");Serial.print(Bufferflushcross/1000);Serial.print(" s");

 Serial.print("  STEP 4:: ");Serial.print("Buffer flushes Capillary:
");Serial.print(Bufferflushcap/1000);Serial.print(" s");

 Serial.println();

 Serial.print("STEP 5:: ");Serial.print("  Buffer pressure release:
");Serial.print(BufferPress/1000);Serial.print(" s");

 Serial.print("  STEP 6:: ");Serial.print("Buffer to waste ");

 Serial.print("  STEP 7:: ");Serial.print("Analyte flushes crosspiece:
");Serial.print(Analyteflushcross/1000);Serial.print(" s");

 Serial.println();

 Serial.print("STEP 8:: ");Serial.print("Analyte is injected:
");Serial.print(Injectiontime);Serial.print(" s");Serial.print("s ");Serial.print("PWM =
");Serial.print(pwm);

 Serial.print("  STEP 9:: ");Serial.print("Analyte pressure release:
");Serial.print(AnalytePress/1000);Serial.print(" s");

 Serial.println();

 Serial.print("STEP 10:: ");Serial.print("Analyte to Waste; Buffer through crosspiece:
");Serial.print(BufferClearCross/1000);Serial.print(" s   ");

 Serial.print("  STEP 11:: ");Serial.print("Isolate Analyte plug with Buffer:
");Serial.print(Plugtime/1000);Serial.print(" s");


  Serial.println();

  Serial.println();


  float TimeStart = millis();

  float TimeNow = millis();

  while ((TimeNow - TimeStart) < TimeSepn)

   {
```

```
  TimeNow = millis();

 float sensorValue = analogRead(A0);
 float sum=0;
 for (int j=0; j<OversampleNum;j++)
 {
  sum += analogRead(A0);
 }

Serial.print(((TimeNow - TimeStart)/1000),3);
Serial.print(" , ");
float AvgVolt = 0.00132927*(sum+14.34126122);
Serial.println(AvgVolt,4);

 }

   Serial.print("END OF RUN NUMBER "); Serial.print(k-1);
   Serial.println();
  // Serial.println(" STEP 11 C0MPLETE");

/*STEP 12: RE-SET PARAMETERS AND RETURN TO
THE "do" loop JUST BEFORE STEP 3. */

//digitalWrite(PumpBuffer,HIGH);
digitalWrite(SolenoidBuffer,LOW);
analogWrite(PumpAnalyte,180);
 digitalWrite(SolenoidAnalyte,LOW);
//digitalWrite(PumpEndCap,HIGH);
digitalWrite(SolenoidCtrl,LOW);
digitalWrite(HighVolt,LOW);
digitalWrite(SensorLight,LOW);
```

```
analogReference(DEFAULT);
Serial.println("STEP 12 C0MPLETE");
Serial.println();
}
while(k<Setnumber);
//NOW TURN EVERYTHING OFF, WAIT FOR "TimeBetweenSets" SECONDS
//AND RETURN TO BEGINNING

digitalWrite(PumpBuffer,LOW);
digitalWrite(PumpAnalyte,LOW);
digitalWrite(SolenoidBuffer,LOW);
//digitalWrite(SolenoidAnalyte,LOW);
digitalWrite(SolenoidCtrl,LOW);
//digitalWrite(HighvoltOne,LOW);
//digitalWrite(HighvoltTwo,LOW);
//digitalWrite(SensorLight,LOW);

Serial.println("  RETURNING TO STEP 1 FOR NEXT SET OF RESULTS...");

delay(TimeBetweenSets);

}
```

APPENDIX 3

Table Apx 3.1: Movement of algorithm performance with chosen index depending on which combinations of criteria are used. This analysis was extracted from the raw data of performances shown in Chapter 3.

Algorithm Order by SNR + Wp	SNR + Wp Raw Score	Algorithm Order by Resolution	Resolution Raw Score	Final Order SNR+Wp+Res	SNR+Wp+Res Raw Score
Savitsky-Golay 605 point	5.906+6.000	Nyquist 241 point	6.000	Nyquist 241 point	17.845
Nyquist 241 point	6.000+5.845	Savitsky-Golay 605 point	5.373	Savitsky-Golay 605 point	17.279
Gauss 381 point	5.777+5.801	Geom. MS 141 point	4.795	Gauss 381 point	14.375
Exp. MS 183 point	5.880+2.879	Exp.MS 183 point	3.669	Exp.MS 183 point	12.428
Boxcar 211 point	5.892+1.000	Gauss 381 point	2.797	Geom. MS 141 point	11.609
Geom. MS 141 point	1.000+5.814	Boxcar 211 point	1.000	Boxcar 211 point	7.892

Table Apx 3.2: Movement of algorithm performance with chosen index depending on which combinations of criteria are used. This analysis was extracted from the raw data of performances shown in Chapter 3.

SAVITSKY-GOLAY CONVOLUTION INTEGERS

p	h	a_{-12}	a_{-11}	a_{-10}	a_{-9}	a_{-8}	a_{-7}	a_{-6}	a_{-5}	a_{-4}	a_{-3}	a_{-2}	a_{-1}	a_0	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	a_{10}	a_{11}	a_{12}
5	35	0	0	0	0	0	0	0	0	0	0	-3	12	17	12	-3	0	0	0	0	0	0	0	0	0	0
7	21	0	0	0	0	0	0	0	0	0	-2	3	6	7	6	3	-2	0	0	0	0	0	0	0	0	0
9	231	0	0	0	0	0	0	0	0	-21	14	39	54	59	54	39	14	-21	0	0	0	0	0	0	0	0
11	429	0	0	0	0	0	0	0	-36	9	44	69	84	89	84	69	44	9	-36	0	0	0	0	0	0	0
13	143	0	0	0	0	0	0	-11	0	9	16	21	24	25	24	21	16	9	0	-11	0	0	0	0	0	0
15	1105	0	0	0	0	0	-78	-13	42	87	122	147	162	167	162	147	122	87	42	-13	-78	0	0	0	0	0
17	323	0	0	0	0	-21	-6	7	18	27	34	39	42	43	42	39	34	27	18	7	-6	-21	0	0	0	0
19	2261	0	0	0	-136	-51	24	89	144	189	224	249	264	269	264	249	224	189	144	89	24	-51	-136	0	0	0
21	3059	0	0	-171	-76	9	84	149	204	249	284	309	324	329	324	309	284	249	204	149	84	9	-76	-171	0	0
23	805	0	-42	-21	-2	15	30	43	54	63	70	75	78	79	78	75	70	63	54	43	30	15	-2	-21	-42	0
25	5175	-253	-138	-33	62	147	222	287	343	387	422	447	462	467	462	447	422	387	343	287	222	147	62	-33	-138	-253

APPENDIX 4

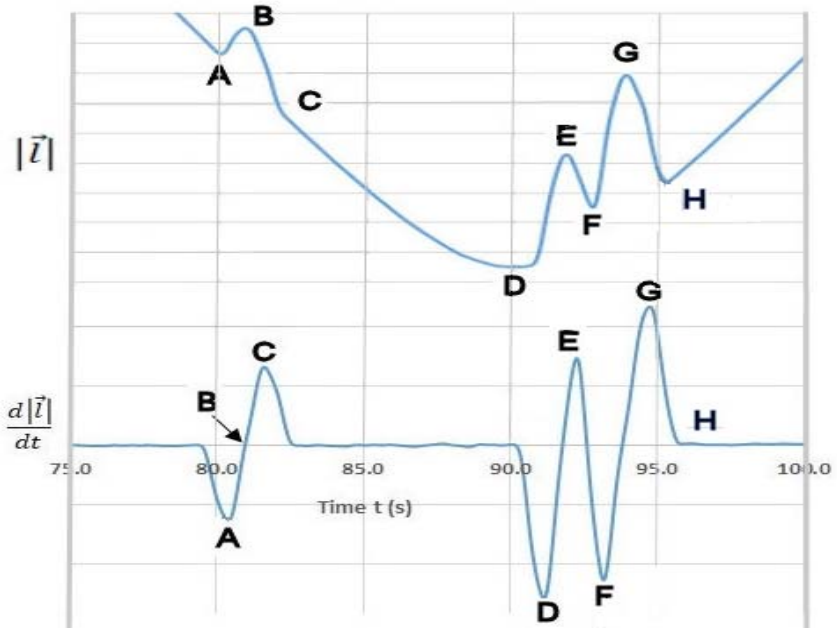


Figure Apx 4: Taken from extracted segment of interest (75.0 s to 100.0 s) in the smoothed electropherogram SG605(G381) of separation of C-102+F+FITC shown in Figure 4.3.2 in Chapter 4.

Clarification: In the plot of $|\vec{l}|$ ~ t the overall appearance is essentially the same as that of the original electropherogram superimposed on a quadratic parabola which is a consequence of $|\vec{l}|$ changing length whilst being of fixed locus at an anchor point.

What it does is highlight the inflection points at A, C, D, F, and H. The points C and D are very hard to identify unambiguously by visual inspection, but he function $\frac{d|\vec{l}|}{dt}$ is able to do this by showing where the gradient changes direction at A, C, D, E, F, and G. Zero points such as B indicate the maxima of each peak.

APPENDIX 5

	PEAK 1	ROW No.	PEAK 2	ROW No.	PEAK 3	ROW No.
Value =	789.642130		789.644107		789.560112	
Time =	87.414	524495	112.549	675307	135.411	810679
Time Start	86.768	520620	111.816	670907	134.730	808392
Time End	88.176	529066	113.251	679515	136.376	818270
Width	**1.408**		**1.435**		**1.646**	
Base Start	789.038877		789.182010		789.296466	
Base End	789.087386		789.139534		789.253569	
Base Ave	**789.063132**		**789.160772**		**789.275018**	
Height	**0.578998**		**0.483335**		**0.285094**	
Noise 1	0.075230035	506051		658529		794267
Noise 2		550055	0.090667687	700191		837821
Noise 3					0.09451259	
SNR=	**7.70**		**5.33**		**3.02**	

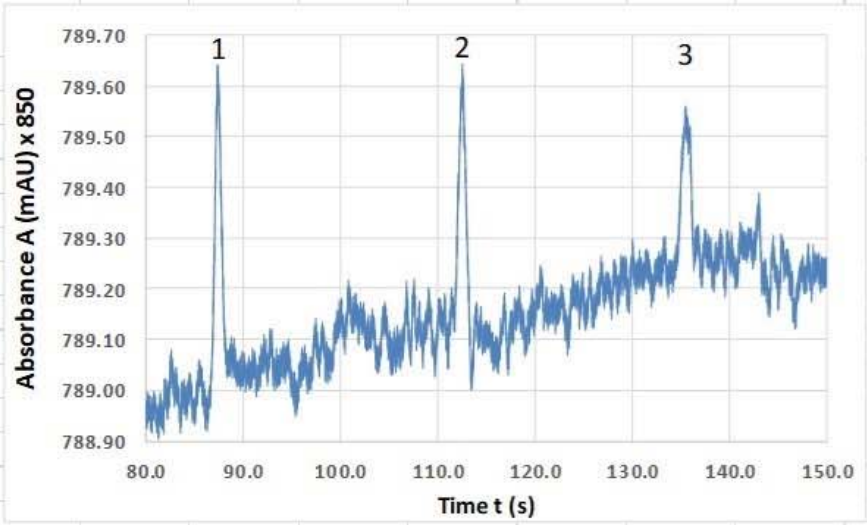
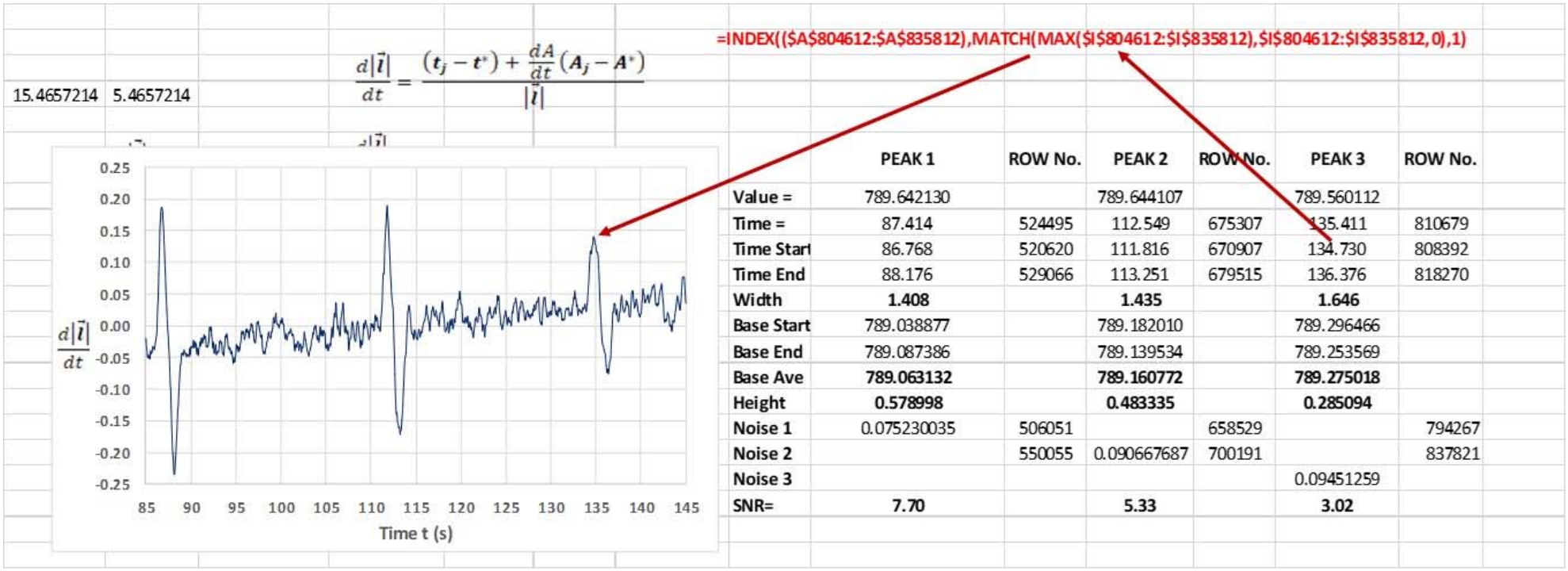


Figure Apx 5.1: The layout of the spreadsheet cells to determine peak absorbance value, peak width, baseline, peak height and SNR.



	PEAK 1	ROW No.	PEAK 2	ROW No.	PEAK 3	ROW No.
Value =	789.642130		789.644107		789.560112	
Time =	87.414	524495	112.549	675307	135.411	810679
Time Start	86.768	520620	111.816	670907	134.730	808392
Time End	88.176	529066	113.251	679515	136.376	818270
Width	**1.408**		**1.435**		**1.646**	
Base Start	789.038877		789.182010		789.296466	
Base End	789.087386		789.139534		789.253569	
Base Ave	**789.063132**		**789.160772**		**789.275018**	
Height	**0.578998**		**0.483335**		**0.285094**	
Noise 1	0.075230035	506051		658529		794267
Noise 2		550055	0.090667687	700191		837821
Noise 3					0.09451259	
SNR=	**7.70**		**5.33**		**3.02**	

Figure Apx 5.2: Illustration of the use of the INDEX and MATCH function to determine the values of t_{left} and t_{right} from the derivative of the length vector to determine peak width.

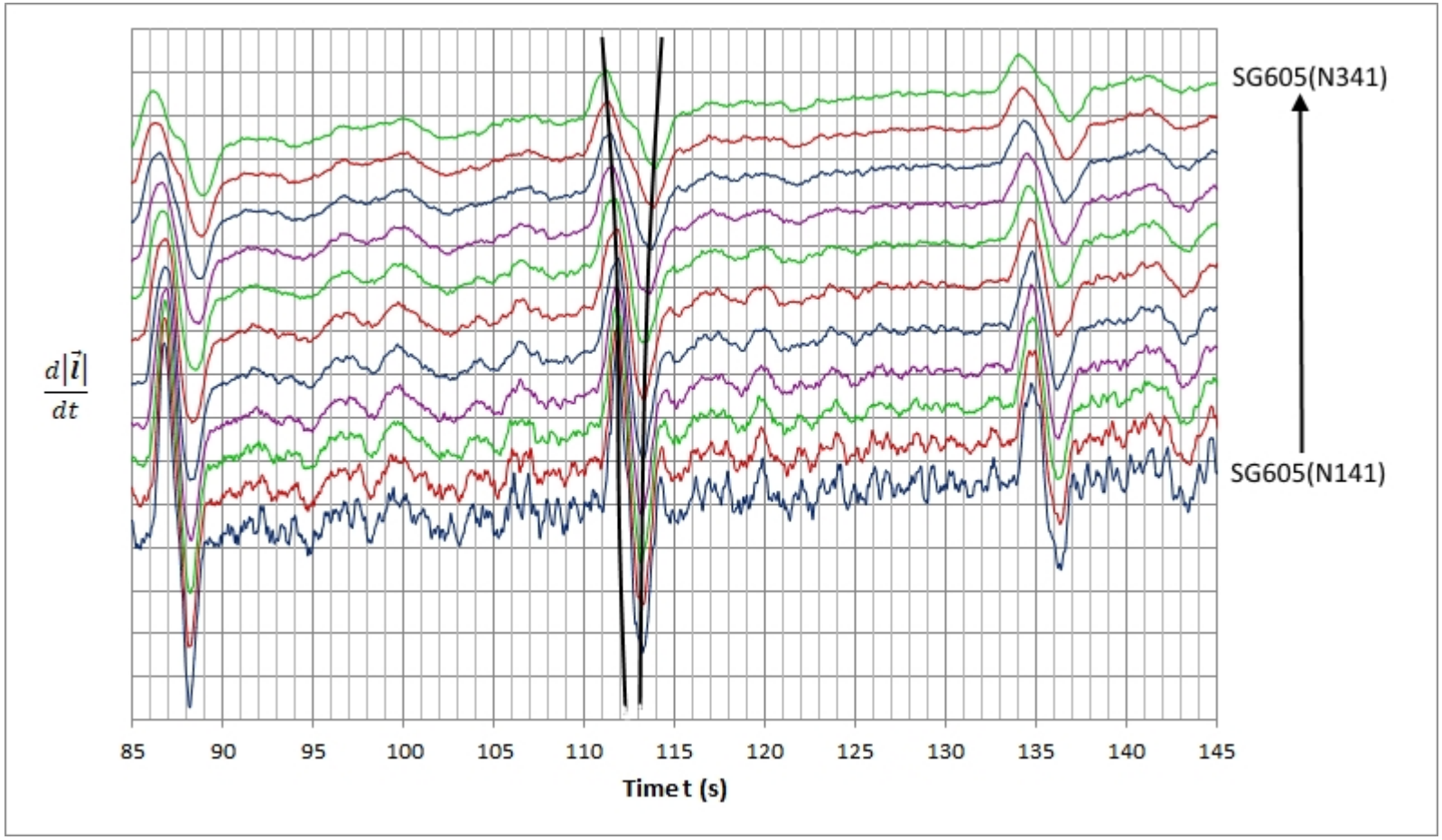


Figure Apx 5.3: Illustration of the use of the vector length derivative to determine peak width. The black lines at Peak 2 show divergence, which measures the increase in peak width as the window size of the internal algorithm increases.

What follows here is a tabular outline of the embedding of the Gaussian algorithm inside the optimised 605-point Savitsky-Golay algorithm. This is followed by the static coding of SG605(G281-481)

SAVITSKY-GOLAY 605

$$S(t) = \left(\frac{1}{\sum_{i=-k}^{k} C_i}\right)\left[C_0 A_0 + \sum_{i=1}^{\frac{k+1}{2}} C_i (A_{i-1} + A_{i+1})\right]$$

For a 25-point polynomial, the central point is point No. 13. A 25-point polynomial fits into 605 point smoothing window as follows:

POLY	$\sum(C_i)$																								
25	5175	-253	-138	-33	62	147	222	287	343	387	422	447	462	**467**	462	447	422	387	343	287	222	147	62	-33	-138	-253
PTS		1-24	25-49	50-74	75-98	99-123	124-147	148-171	172-195	196-219	220-243	244-267	268-302	**303**	304-328	329-352	353-376	377-410	411-434	435-458	459-482	483-507	508-531	532-555	556-580	581-605
CELLS	START	12	37	62	87	111	136	160	184	208	232	256	280		316	341	365	389	423	447	471	495	520	544	568	593
	END	36	61	86	110	135	159	183	207	231	255	279	314	**315**	340	364	388	422	446	470	494	519	543	567	592	617
	FORMULA $\left(\frac{1}{5175}\right)\times$																								

Again, it is only possible to nest the smaller window inside the larger one. So Gauss(381) has to nest within S-G(605)

GAUSS 381

D					59					106					154					202	203					250					298					345					393
Gs	12					60					107					155					202		204					251					299					346				
																					203																				
	← 48 →					← 47 →					← 48 →					← 47 →						× 0.5	← 47 →					← 48 →					← 47 →					← 48 →				
	0.001					0.021					0.136					0.341							0.341					0.136					0.021					0.001				
	× 0.5																						× 0.5																				

So the nesting must occur as follows:

25	5175	-253	-138	-33	62	147	222	287	343	387	422	447	462	**467**	462	447	422	387	343	287	222	147	62	-3
PTS		1-24	25-49	50-74	75-98	99-123	124-147	148-171	172-195	196-219	220-243	244-267	268-302	**303**	304-328	329-352	353-376	377-410	411-434	435-458	459-482	483-507	508-531	53 55
CELLS	START	12	37	62	87	111	136	160	184	208	232	256	280		316	341	365	389	423	447	471	495	520	54
	END	36	61	86	110	135	159	183	207	231	255	279	314	**315**	340	364	388	422	446	470	494	519	543	56

		12	37	60	62	87	107	111	136	155	160	184	**203**	204	208	232	251	256	280	299	315	341	346	365	389
		36	59	61	86	106	110	135	154	159	183	202		207	231	250	255	279	298	314	340	345	364	388	393
Weighting		0.001		0.021			0.136			0.341			**0.5**	0.341			0.136			0.021			0.001		

SG605

=(1/5175)*(($I$24)*$B315+($I$23)*(AVERAGE($B280:$B314))+($I$25)*(AVERAGE($B316:$B340))+($I$22)*(AVERAGE($B256:$B279))+($I$26)*(AVERAGE($B341:$B364))+($I$21)*(AVERAGE($B232:$B255))+($I$27)*(AVERAGE($B365:$B388))+($I$20)*(AVERAGE($B208:$B231))+($I$28)*(AVERAGE($B389:$B422))+($I$19)*(AVERAGE($B184:$B207))+($I$29)*(AVERAGE($B423:$B446))+($I$18)*(AVERAGE($B160:$B183))+($I$30)*(AVERAGE($B447:$B470))+($I$17)*(AVERAGE($B136:$B159))+($I$31)*(AVERAGE($B471:$B494))+($I$16)*(AVERAGE($B111:$B135))+($I$32)*(AVERAGE($B495:$B519))+($I$15)*(AVERAGE($B87:$B110))+($I$33)*(AVERAGE($B520:$B543))+($I$14)*(AVERAGE($B37:$B61))+($I$34)*(AVERAGE($B544:$B567))+($I$13)*(AVERAGE($B37:$B61))+($I$35)*(AVERAGE($B568:$B592))+($I$12)*(AVERAGE($B12:$B36))+($I$36)*(AVERAGE($B593:$B617)))

GAUSS 381

=$I$11*(0.5*AVERAGE($B12:$B393) + 0.5*0.341*SUM($B155:$B202)+ 0.5*0.341*SUM($B204:$B250)+0.5*0.136*SUM($B107:$B154)+ 0.5*0.136*SUM($B251:$B298)+0.5*0.021*SUM($B60:$B106)+ 0.5*0.021*SUM($B299:$B345)+0.5*0.001*SUM($B12:$B59)+ 0.5*0.001*SUM($B346:$B393)))

SG605(GAUSS381)

UNWEIGHTED

=(1/5175)*(($I$12*AVERAGE($B12:$B36)+$I$11*0.0005*SUM($B12:$B36))+($I$13*AVERAGE($B37:$B59)+$I$11*0.0005*SUM($B37:$B59))+($I$13*AVERAGE($B60:$B61)+$I$11*0.0105*SUM($B60:$B61))+($I$14*AVERAGE($B62:$B86)+$I$11*0.0105*SUM($B62:$B86))+($I$15*AVERAGE($B87:$B106)+$I$11*0.0105*SUM($B87:$B106))+($I$15*AVERAGE($B107:$B110)+$I$11*0.068*SUM($B107:$B110))+($I$16*AVERAGE($B111:$B135)+$I$11*0.068*SUM($B111:$B135))+($I$17*AVERAGE($B136:$B154)+$I$11*0.068*SUM($B136:$B154))+($I$17*(AVERAGE($B155:$B159)+$I$11*0.1705*SUM($B155:$B159))+($I$18*AVERAGE($B160:$B183)+$I$11*0.1705*SUM($B160:$B183))+($I$19*AVERAGE($B184:$B202)+$I$11*0.1705*SUM($B184:$B202))+($I$19*$B203+$I$11*0.5*AVERAGE($B12:$B393))+($I$19*AVERAGE($B204:$B207)+$I$11*0.1705*SUM($B204:$B207))+($I$20*AVERAGE($B208:$B231)+$I$11*0.1705*SUM($B208:$B231))+($I$21*AVERAGE($B232:$B250)+$I$11*0.1705*SUM($B232:$B250))+($I$21*AVERAGE($B251:$B255)+$I$11*0.068*SUM($B251:$B255))+($I$22*AVERAGE($B256:$B279)+$I$11*0.068*SUM($B256:$B279))+($I$23*AVERAGE($B280:$B298)+$I$11*0.068*SUM($B280:$B298))+($I$23*AVERAGE($B299:$B314)+$I$11*0.0105*SUM($B299:$B314))+($I$24*$B315+$I$11*0.0105*$B315)+($I$25*AVERAGE($B316:$B340)+$I$11*0.0105*SUM($B316:$B340))+($I$26*AVERAGE($B341:$B345)+$I$11*0.0105*SUM($B341:$B345))+($I$26*AVERAGE($B346:$B364)+$I$11*0.00005*SUM($B346:$B364))+($I$27*AVERAGE($B365:$B388)+$I$11*0.00005*SUM($B365:$B388))+($I$28*AVERAGE($B389:$B393)+$I$11*0.00005*SUM($B389:$B393))+($I$28*AVERAGE($B394:$B422))+($I$29*AVERAGE($B423:$B446))+($I$30*AVERAGE($B447:$B470))+($I$31*AVERAGE($B471:$B494))+($I$32*AVERAGE($B495:$B519))+($I$33*AVERAGE($B520:$B543))+($I$34*AVERAGE($B544:$B567))+($I$35*AVERAGE($B568:$B592))+($I$36*AVERAGE($B593:$B617))))

WEIGHTED

=(1/5175)*(((1/26)*$I$12*AVERAGE($B12:$B36)+ (25/26)*$I$11*0.0005*SUM($B12:$B36))+((1/24)*$I$13*AVERAGE($B37:$B59)+
(23/24)*$I$11*0.0005*SUM($B37:$B59))+((1/3)*$I$13*AVERAGE($B60:$B61)+ (2/3)*$I$11*0.0105*SUM($B60:$B61))+(
(1/26)*$I$14*AVERAGE($B62:$B86)+ (25/26)*$I$11*0.0105*SUM($B62:$B86))+((1/21)*$I$15*AVERAGE($B87:$B106)+
(20/21)*$I$11*0.0105*SUM($B87:$B106))+((1/5)*$I$15*AVERAGE($B107:$B110)+ (4/5)*$I$11*0.068*SUM($B107:$B110))+(
(1/26)*$I$16*AVERAGE($B111:$B135)+ (25/26)*$I$11*0.068*SUM($B111:$B135))+((1/20)*$I$17*AVERAGE($B136:$B154)+
(19/20)*$I$11*0.068*SUM($B136:$B154))+((1/6)*$I$17*(AVERAGE($B155:$B159)+ (5/6)*$I$11*0.1705*SUM($B155:$B159))+(
(1/25)*$I$18*AVERAGE($B160:$B183)+ (24/25)*$I$11*0.1705*SUM($B160:$B183))+((1/20)*$I$19*AVERAGE($B184:$B202)+
(19/20)*$I$11*0.1705*SUM($B184:$B202))+((1/2)*$I$19*$B203+(1/2)*$I$11*0.5*AVERAGE($B12:$B393))+((1/5)*$I$19*AVERAGE($B204:$B207)+
(4/5)*$I$11*0.1705*SUM($B204:$B207))+((1/25)*$I$20*AVERAGE($B208:$B231)+ (24/25)*$I$11*0.1705*SUM($B208:$B231))+(
(1/20)*$I$21*AVERAGE($B232:$B250)+ (19/20)*$I$11*0.1705*SUM($B232:$B250))+((1/6)*$I$21*AVERAGE($B251:$B255)+
(5/6)*$I$11*0.068*SUM($B251:$B255))+((1/25)*$I$22*AVERAGE($B256:$B279)+ (24/25)*$I$11*0.068*SUM($B256:$B279))+(
(1/20)*$I$23*AVERAGE($B280:$B298)+ (19/20)*$I$11*0.068*SUM($B280:$B298)) +((1/17)*$I$23*AVERAGE($B299:$B314)+
(16/17)*$I$11*0.0105*SUM($B299:$B314))+((1/2)*$I$24*$B315+(1/2)*$I$11*0.0105*$B315)+((1/26)*$I$25*AVERAGE($B316:$B340)+
(25/26)*$I$11*0.0105*SUM($B316:$B340))+((1/6)*$I$26*AVERAGE($B341:$B345)+ (5/6)*$I$11*0.0105*SUM($B341:$B345))+(
(1/20)*$I$26*AVERAGE($B346:$B364)+ (19/20)*$I$11*0.00005*SUM($B346:$B364))+((1/25)*$I$27*AVERAGE($B365:$B388)+
(24/25)*$I$11*0.00005*SUM($B365:$B388))+((1/6)*$I$28*AVERAGE($B389:$B393)+
(5/6)*$I$11*0.00005*SUM($B389:$B393))+($I$28*AVERAGE($B394:$B422))+($I$29*AVERAGE($B423:$B446))+($I$30*AVERAGE($B447:$B470))+($I$31*AV
ERAGE($B471:$B494))+($I$32*AVERAGE($B495:$B519))+($I$33*AVERAGE($B520:$B543))+($I$34*AVERAGE($B544:$B567))+($I$35*AVERAGE($B568:$B592
))+($I$36*AVERAGE($B593:$B617))))